SUMEX

STANFORD UNIVERSITY
MEDICAL EXPERIMENTAL COMPUTER RESOURCE

RR-00785

COMPETING RENEWAL APPLICATION

Submitted to

BIOMEDICAL RESEARCH TECHNOLOGY PROGRAM
NATIONAL INSTITUTES OF HEALTH

June 1, 1985

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

Form approved
OMB No. 0925-0001

DEPARTMENT OF HEALTH AND HUMAN SERVICES
PUBLIC HEALTH SERVICE

GRANT APPLICATION

FOLLOW INSTRUCTIONS CAREFULLY

LEAVE BLANK		
TYPE	ACTIVITY	NUMBER
REVIEW GROUP		FORMERLY
COUNCIL/BOARD *(Month, year)*		DATE RECEIVED

1. TITLE OF APPLICATION *(Do not exceed 56 typewriter spaces)*
SU Medical Experimental Computer Resource (SUMEX)

2. RESPONSE TO SPECIFIC PROGRAM ANNOUNCEMENT ☐ NO ☐ YES *(If "YES," state RFA number and/or announcement title)*

3. PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR

3a. NAME *(Last, first, middle)*
Shortliffe, Edward H.

3b. SOCIAL SECURITY NUMBER
[redacted]

3c. POSITION TITLE
Associate Professor of Medicine

3d. MAILING ADDRESS *(Street, city, state, zip code)*
Stanford University School of Medicine
Room TB-105
Stanford, CA 94305

3e. DEPARTMENT, SERVICE, LABORATORY OR EQUIVALENT
Department of Medicine

3f. MAJOR SUBDIVISION
School of Medicine

3g. TELEPHONE *(Area code, number and extension)*
(415)497-6979

4. HUMAN SUBJECTS
☒ NO ☐ YES { ☐ Exemption # ________ OR ☐ Form HHS 596 enclosed }

5. RECOMBINANT DNA
☒ NO ☐ YES

6. DATES OF ENTIRE PROPOSED PROJECT PERIOD
From: 8/1/86 Through: 7/31/91

7. DIRECT COSTS REQUESTED FOR FIRST 12-MONTH BUDGET PERIOD *(from page 4)*
$ 1,370,257

8. DIRECT COSTS REQUESTED FOR ENTIRE PROPOSED PROJECT PERIOD *(from page 5)*
$ 6,963,247

9. PERFORMANCE SITES *(Organizations and addresses)*
Stanford University
School of Medicine
Stanford, CA 94305

10. INVENTIONS *(Competing continuation application only)*
☒ NO ☐ YES { ☐ Previously reported OR ☐ Not previously reported }

11. APPLICANT ORGANIZATION *(Name, address, and congressional district)*
Stanford University
c/o Sponsored Projects Office
Encina Hall, Room 40
Stanford, CA 94305
Congressional District No. 11

12. TYPE OF ORGANIZATION
☐ Public. Specify ☐ Federal ☐ State ☐ Local
☒ Private Nonprofit
☐ For Profit *(General)*
☐ For Profit *(Small Business)*

13. ENTITY IDENTIFICATION NUMBER
IRS No. 94-1156365

14. ORGANIZATIONAL COMPONENT TO RECEIVE CREDIT FOR BIOMEDICAL RESEARCH SUPPORT GRANT
Code 0 1 Description

15. OFFICIAL IN BUSINESS OFFICE TO BE NOTIFIED IF AN AWARD IS MADE *(Name, title, address and telephone number.)*
Patricia Byers, Contract Officer
Sponsored Projects Office
Encina Hall, Room 40
Stanford, CA 94305
(415)497-2883

16. OFFICIAL SIGNING FOR APPLICANT ORGANIZATION *(Name, title, address and telephone number)*
PATRICIA BYERS SENIOR CONTRACT OFFICER
Sponsored Projects Office
Encina Hall, Room 40
Stanford, CA 94305
(415)497-2883

17. PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR ASSURANCE: I agree to accept responsibility for the scientific conduct of the project and to provide the required progress reports if a grant is awarded as a result of this application. Willful provision of false information is a criminal offense *(U.S. Code, Title 18, Section 1001)*.

SIGNATURE OF PERSON NAMED IN 3a *(In ink. "Per" signature not acceptable)*
Edward H Shortliffe
DATE 5/24/85

18. CERTIFICATION AND ACCEPTANCE: I certify that the statements herein are true and complete to the best of my knowledge, and accept the obligation to comply with Public Health Service terms and conditions if a grant is awarded as the result of this application. A willfully false certification is a criminal offense *(U.S. Code, Title 18, Section 1001)*.

SIGNATURE OF PERSON NAMED IN 16 *(In ink. "Per" signature not acceptable)*
Patricia Byers
DATE 5/29/85

E. Shortliffe	Principal Investigator	Medicine & Computer Science
E. Feigenbaum	Co-Principal Investigator	Computer Science
T. Rindfleisch	Director of Resource	Medicine & Computer Science
C. Jacobs	ONCOCIN Investigator	Medicine
L. Fagan	AIM Liaison	Medicine
W. Yeager	Systems Programmer/Assistant Resource Director	Medicine
B. Buchanan	Professor of Research - Computer Science	Computer Science
B. Hayes-Roth	Senior Research Associate	Computer Science
H. Brown	Senior Research Associate	Computer Science
P. Nii	Research Associate	Computer Science

ABSTRACT OF RESEARCH PLAN: State the application's long-term objectives and specific aims, making reference to the health relatedness of the project, and describe concisely the methodology for achieving these goals. Avoid summaries of past accomplishments and the use of the first person. The abstract is meant to serve as a succinct and accurate description of the proposed work when separated from the application. DO NOT EXCEED THE SPACE PROVIDED.

Stanford University is developing and operating a national shared computing resource (SUMEX-AIM), in partnership with the NIH Biomedical Research Technology Program, to explore applications of computer science research in artificial intelligence (AI) to health research. There are three main objectives of the resource: 1) to develop and provide the computing resources and human assistance needed by scientists working on a broad range of biomedical applications of AI; 2) to demonstrate that it is feasible to provide resources and assistance to a national community of researchers, integrating distributed and centralized computing technology with local and national computer communication networks; and 3) to develop the community of scientists interested in working on AI in Medicine (AIM), promoting its growth and vigor through electronic communications. Besides the economic advantages of resource sharing made possible by electronic communications, we believe that a new style of science is emerging from communications-enhanced settings.

AI research is aimed at understanding the principles of computer-based symbolic knowledge representation, reasoning, and problem-solving processes and applying these to increase the computer's effectiveness as a tool in knowledge-intensive fields like medicine and biology. Our research work is driven by real-world scientific applications, chosen because of their relevance to current important biomedical problems and because they expose key underlying AI research issues. Current application areas include programs for differential diagnosis, cancer chemotherapy protocol management, protein structure inference, and drug interaction advice. Resource core research goals include basic research in areas such as blackboard problem-solving architectures and knowledge acquisition; methodologies for clinical decision-making advisors; and the development of network-based Lisp workstation computing environments.

Additional resource users will be selected within available resource capacity with the help of an AIM Executive Committee and Advisory Group on the basis of reviews of the proposed research. Selection criteria will include general scientific interest and merit, relevance to the resource AI mission, and the community orientation of the collaborator.

X

Table of Contents

List of Figures

1. Budget, Biographies, and Environment

1.1. Total Resource Budget

This section details the Total Resource Budget starting with the first renewal year (resource year 14) beginning August 1, 1986.

TOTAL RESOURCE BUDGET

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

DETAILED BUDGET FOR FIRST 12 MONTH BUDGET PERIOD DIRECT COSTS ONLY

FROM 8/1/86 THROUGH 7/31/87

DOLLAR AMOUNT REQUESTED (Omit cents)

PERSONNEL (Applicant organization only) NAME	POSITION TITLE	TIME/EFFORT %	Hours per Week	SALARY	FRINGE BENEFITS	TOTALS
	Principal Investigator					
see attached sheet						
	SUBTOTALS →			660,335	168,940	829,275

CONSULTANT COSTS

EQUIPMENT (Itemize)

Resource host and network equipment	$14,000	
Experimental Lisp Machines	$75,000	
		89,000

SUPPLIES (Itemize by category)

Office supplies	4,350	
Computer supplies	4,250	
Engineering supplies	7,500	
		16,100

TRAVEL	DOMESTIC	9,500	9,500
	FOREIGN		
PATIENT CARE COSTS	INPATIENT		
	OUTPATIENT		

ALTERATIONS AND RENOVATIONS (Itemize by category)

CONSORTIUM/CONTRACTUAL COSTS

OTHER EXPENSES (Itemize by category) File server maintenance: $19,200; Terminal maintenance: $1,350; Lisp Machine maintenance: $30,000; Misc. software: $1,800; Aux. computing services: $3,000; Documentation: $1,000; Books/publications: $2,650; Office telephones: $13,100; Dataphone lines: $4,000; Repro/services: $2,700; Prorated 2060 Opns Costs (80%): $347,582 — 426,382

TOTAL DIRECT COSTS (Also enter on page 1, item 7) → 1,370,257

First Year Personnel Detail

		%	SALARY	BEN	TOTAL
RESOURCE MANAGEMENT					
E. Shortliffe	Principal Invest.	15			
E. Feigenbaum	Co-Principal Inv.	10			
T. Rindfleisch	Resource Director	70			
L. Fagan	AIM Liaison/ONC Proj. Mgr.	25			
W. Yeager	Asst. Resource Dir.	90			
P. McCabe	Administrator	75			
M. Timothy	Secretary	100			
Open	Receptionist	75			
CORE SYSTEM DEVELOPMENT					
A. Sweer	Systems Pgmr.	10			
F. Gilmurray	Systems Pgmr.	70			
W. Croft	Systems Pgmr.	100			
R. Acuff	Systems Pgmr.	60			
C. Schmidt	Systems Pgmr.	60			
N. Veizades	Electronics Engr.	40			
I. Torres	Engr. Aid	40			
CORE BASIC AI RESEARCH					
B. Buchanan	Professor of Comp. Sci.	10			
B. Hayes-Roth	Sr. Res. Assoc.	15			
H. Brown	Sr. Res. Assoc.	10			
P. Nii	Res. Assoc.	10			
M. Hewett	Programmer	40			
P. Karp	Res. Asst.	62			
A. Garvey	Res. Asst.	62			
J. Brugge	Res. Asst.	62			
CORE ONCOCIN RESEARCH					
C. Jacobs	ONCOCIN Investigator	5			
R. Lenon	Clinical Spec.	25			
C. Lane	Systems Pgmr.	60			
S. Tu	Sci. Programmer	50			
D. Combs	Sci. Programmer	50			
D. Vian	Administrator	25			
J. Rohn	Data Mgr.	100			
A. Grant	Secretary	50			
T. Barsalou	Res. Asst.	62			
L. Perreault	Res. Asst.	62			
SYSTEM OPERATIONS SUPPORT					
R. Tucker	Opns. Mgr.	20			
P. Ryalls	System Mgr.	20			
	SUBTOTAL DIRECT SALARIES		660335		
	STAFF BENEFITS		168940		
	TOTAL OF PERSONNEL		829275		

 PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

BUDGET FOR ENTIRE PROPOSED PROJECT PERIOD
DIRECT COSTS ONLY

BUDGET CATEGORY TOTALS		1st BUDGET PERIOD *(from page 4)*	ADDITIONAL YEARS SUPPORT REQUESTED			
			2nd	3rd	4th	5th
PERSONNEL *(Salary and fringe benefits.) (Applicant organization only)*		829,275	891,340	989,154	1,063,465	1,143,264
CONSULTANT COSTS						
EQUIPMENT		89,000	95,230	101,897	109,029	116,661
SUPPLIES		16,100	17,228	18,433	19,723	21,104
TRAVEL	DOMESTIC	9,500	10,165	10,877	11,638	12,453
	FOREIGN					
PATIENT CARE COSTS	INPATIENT					
	OUTPATIENT					
ALTERATIONS AND RENOVATIONS						
CONSORTIUM/ CONTRACTUAL COSTS						
OTHER EXPENSES		78,800	84,317	90,219	96,533	103,290
Prorated 2060 Opns		347,582	279,616	200,050	107,304	-0-
TOTAL DIRECT COSTS		1,370,257	1,377,896	1,410,630	1,407,692	1,396,772

TOTAL FOR ENTIRE PROPOSED PROJECT PERIOD *(Also enter on page 1, item 8)* → $ 6,963,247

JUSTIFICATION (Use continuation pages if necessary): Describe the specific functions of the personnel and consultants. If a recurring annual increas in personnel costs is anticipated, give the percentage. For *all* years, justify any costs for which the need may not be obvious, such as equipment, foreig travel, alterations and renovations, and consortium/contractual costs. For any additional years of support requested, justify any significant increases i any category over the first 12 month budget period. In addition, for COMPETING CONTINUATION applications, justify any significant increases ove the current level of support.

1.2. 2060 Operations Budget

The budget in this section is for the projected operations costs of the 2060 mainframe system that has been the main resource for national and local users. We will be phasing this link with the past out over the 5-year term of this grant in favor of the new distributed workstation environment we plan to develop. As the first step in the phase-out, we have included 80% of the first-year 2060 operating costs in the first-year Total Resource Budget above. In future years, we include proportionately less of these costs, reducing the pro rata share by 20% per year.

2060 OPERATING BUDGET

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

DETAILED BUDGET FOR FIRST 12 MONTH BUDGET PERIOD DIRECT COSTS ONLY

FROM 8/1/86 THROUGH 7/31/87

DOLLAR AMOUNT REQUESTED (Omit cents)

PERSONNEL (Applicant organization only) NAME	POSITION TITLE	TIME/EFFORT %	Hours per Week	SALARY	FRINGE BENEFITS	TOTALS
	Principal Investigator					
see attached sheet						
	SUBTOTALS →			186,268	47,655	233,923

Category	Item	Amount	TOTALS
CONSULTANT COSTS			
EQUIPMENT (Itemize)	2060 Accessories and Equipment	6,000	6,000
SUPPLIES (Itemize by category)	Office supplies	920	
	Computer supplies	8,000	
	Engineering supplies	1,500	10,420
TRAVEL	DOMESTIC	1,500	1,500
	FOREIGN		
PATIENT CARE COSTS	INPATIENT		
	OUTPATIENT		
ALTERATIONS AND RENOVATIONS (Itemize by category)			
CONSORTIUM/CONTRACTUAL COSTS			
OTHER EXPENSES (Itemize by category)	2060 maintenance: $92,300; DEC software maintenance: $3,950; Software licenses: $6,800; Documentation: $1,200; Books/publications: $625; Office telephones: $2,935; Dataphone lines: $14,000; Repro/services: $825; TYMNET network services: $60,000		182,635
TOTAL DIRECT COSTS (Also enter on page 1, item 7) →			$434,478

First Year Personnel Detail for 2060 Operations

		%	SALARY	BEN	TOTAL
MANAGEMENT					
T. Rindfleisch	Resource Director	10			
W. Yeager	Asst. Resource Dir.	10			
P. McCabe	Administrator	25			
SYSTEM STAFF					
A. Sweer	Systems Pgmr.	90			
F. Gilmurray	Systems Pgmr.	30			
ELECTRONICS STAFF					
N. Veizades	Electronics Engr.	20			
I. Torres	Engr. Aid	20			
OPERATIONS SUPPORT					
R. Tucker	Opns. Mgr.	80			
P. Ryalls	System Mgr.	80			
M. Blattel	Student Oper.	47			
N. Dolhert	Student Oper.	22			
A. Jong	Student Oper.	22			
	SUBTOTAL DIRECT SALARIES		186268		
	STAFF BENEFITS		47655		
	TOTAL OF PERSONNEL		233923		

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

BUDGET FOR ENTIRE PROPOSED PROJECT PERIOD DIRECT COSTS ONLY

BUDGET CATEGORY TOTALS		1st BUDGET PERIOD *(from page 4)*	ADDITIONAL YEARS SUPPORT REQUESTED			
			2nd	3rd	4th	5th
PERSONNEL *(Salary and fringe benefits.) (Applicant organization only)*		233,923	251,433	270,510	290,830	312,656
CONSULTANT COSTS						
EQUIPMENT		6,000	6,420	6,869	7,350	7,865
SUPPLIES		10,420	11,149	11,929	12,765	13,658
TRAVEL	DOMESTIC	1,500	1,605	1,717	1,838	1,966
	FOREIGN					
PATIENT CARE COSTS	INPATIENT					
	OUTPATIENT					
ALTERATIONS AND RENOVATIONS						
CONSORTIUM/ CONTRACTUAL COSTS						
OTHER EXPENSES		182,635	195,420	209,099	223,737	239,396
TOTAL DIRECT COSTS		434,478	466,027	500,124	536,520	575,541

TOTAL FOR ENTIRE PROPOSED PROJECT PERIOD *(Also enter on page 1, item 8)* → $ 2,512,690

JUSTIFICATION (Use continuation pages if necessary): Describe the specific functions of the personnel and consultants. If a recurring annual increas in personnel costs is anticipated, give the percentage. For *all* years, justify any costs for which the need may not be obvious, such as equipment, foreig travel, alterations and renovations, and consortium/contractual costs. For any additional years of support requested, justify any significant increases any category over the first 12 month budget period. In addition, for COMPETING CONTINUATION applications, justify any significant increases ov the current level of support.

Of these costs, the following are included in the resource budget:

Year 1: $347,582 (80%)
Year 2: $279,616 (60%)
Year 3: $200,050 (40%)
Year 4: $107,304 (20%)
Year 5: -0- (0%)

1.3. Budget Explanation and Justification

1.3.1. Total Resource Budget

This section explains the details of our resource budget plan over the proposed five year grant term, including both the SUMEX renewal and the merged ONCOCIN Dissemination Studies core research (see page 53). Details of the 2060 operations costs are explained in the next section.

In overview, this budget covers a portion of the resource core research and management costs, basic workstation and network environment operations costs and a prorated share of the mainframe computing facility operations costs for the local and national communities. Reviewers will note that only portions of most resource staff members are charged to this budget, the remaining salary support coming from other funding for individual core research and collaborative projects (see page 105). Also, the proposed funding for experimental Lisp machine hardware is a small fraction of the total workstation hardware investment already in place from support received from NIH, DARPA, ONR, and industrial gifts. As a benchmark of the relative magnitude (and hence leverage) of the proposed funding for this resource grant, as compared to other sources of support for this work, consider a snapshot of the year 1 budget. The proposed $1.37M direct cost funding translates to approximately $2.25M in total costs (including indirect costs) as compared to well over $6M in annual total cost funding for KSL work at Stanford. This does not include estimates of the funding base for non-Stanford collaborative users of the resource. It should be emphasized though that this DRR support of the SUMEX-AIM computing resource has been and remains an essential enabling complement to the other sources of support and makes possible the overall scope of our work.

Reviewers will also note that our 5-year budget is essentially flat, despite the inclusion of 7% annual inflation factors. This is because we have linearly phased-out requested DRR support for what has been the mainstay SUMEX-AIM resource, the DEC 2060. In the coming era of workstations, we feel it is important to withdraw support from that part of the resource, but to do so in a responsible fashion that allows time for the national community of projects to find alternative sources of computing support and for core system developments to offer alternatives for our own work and that of the national community. We budget no DRR support for the DEC 2020 demonstration machine or the shared VAX 11/780 time-sharing machine.

Indirect costs are not shown in the budget and will be awarded separately on the basis of Modified Total Direct Costs. The indirect cost rate of 69%, is based on an agreement with the Office of Naval Research (ONR) dated September 14, 1984.

Personnel

The proposed personnel budget is based on current staffing necessary for the proposed work. The estimates are derived from actual salaries for our project staff, including resource management, core research and development, and operations support for collaborative projects. The salary estimates are increased at 7% per year to cover estimated inflation. Staff benefits are computed using the following rates projected by the university for all personnel: 25.4% (9/85-8/86), 25.6% (9/86-8/87), 26.2% (9/87-8/88), 26.9% (9/88-8/89), 27.5% (9/89-8/90) and 28.1% (9/90-8/91).

Resource Management and Overall Technical Direction

Professor Shortliffe (15%) is the resource Principal Investigator, Professor Feigenbaum

(10%) is co-Principal Investigator, and Mr. Rindfleisch (70%) is the Resource Director. All three are responsible for overall resource management and contribute substantially to core research and development efforts as well. Mr. Yeager (90%) is Assistant Resource Director and has responsibility for network and workstation system development. Dr. Fagan (25%[1]) is responsible for liaison with the national AIM community and the AIM management committees and is Manager of the ONCOCIN core research project.

Ms. McCabe (75%) and Ms. Timothy (100%) provide central resource administrative and clerical support for SUMEX and community activities. We plan to hire a receptionist shared between the SUMEX and ONCOCIN/Medical Computer Science groups during the summer of 1985. This person is shown as "Open" and is budgeted at (75%).

Core System Development

The core system development staff, while sharing a substantial joint responsibility for system development, maintenance, user assistance, and operational support, have specific areas of responsibility as follows. Under the direction of Mr. Yeager, already mentioned above, the development of network virtual communications, shared task execution among cooperating workstations, and virtual graphics capabilities will be shared appropriately among staff experts for various relevant environments. In addition, Andy Sweer (10%) and Frank Gilmurray (70%) are responsible for workstation user support and subsystem development such as the merging of text and graphics from various sources and uniform access to printing facilities. William Croft (100%) is responsible for our multiprotocol UNIX file server systems, the development of IP/UDP high-performance file access capabilities, necessary modifications to local area network gateway and interface systems, and network system performance evaluation. Richard Acuff (60%) and Christopher Schmidt (60%) are responsible for Texas Instruments Explorer, Symbolics 3600, and Xerox D-machine support and development. This includes, for example, responsibility of systems support and integration within our Ethernet environment, user support, and vendor liaison. They also are responsible for development of specific system-dependent packages such as electronic mail, text and graphics generation, file management, etc.

Finally, we budget Mr. Nicholas Veizades (40%) as the project electronics engineer and Mr. Israel Torres (40%) his assistant for hardware and maintenance. Mr. Veizades and Mr. Torres are responsible for designing needed special purpose hardware (e.g., communications equipment, intermachine network hardware, and Ethernet interfaces) and for integrating new hardware into the facility, maintaining facility equipment, and correcting communication problems.

Core Basic AI Research

We continue to budget partial support for specific members of the Knowledge Systems Laboratory for core research work to explore basic AI issues relating to biomedical applications and to develop and generalize AI software tools important to the entire SUMEX-AIM community. Prof. Buchanan (10%) will provide managerial and technical direction for staff and students working on proposed core research efforts. Dr. Hayes-Roth (15%) will work on the knowledge-based blackboard control research for the BB1 system which is the tool being used by the PROTEAN project. Dr. Brown (10%) is working on issues of blackboard system design for hierarchical asynchronous

[1]During renewal years 1 and 2, Dr. Fagan is budgeted at only 25%, because part of his salary is supplemented by a New Investigator Award. During years 3-5, when the term of that award ends, he is budgeted at 55%.

concurrency and Ms. Nii (10%) is working of a retrospective of the AGE blackboard system and the ramifications of this control structure for symbolic computing architectures. Mr. Hewett (40%) is a research programmer who will work on knowledge acquisition research. Messrs. Karp, Garvey, and Brugge and graduate Research Assistants who will work on qualitative simulation, learning, and blackboard architecture research respectively.

Core ONCOCIN Dissemination Research

Dr. Charlotte Jacobs (5%) is Co-Principal Investigator on the ONCOCIN Project and is director of the Oncology Clinic at Stanford. She will continue to oversee the clinical implementation of the ONCOCIN workstations in the day-care center. Dr. Rick Lenon (25%), is a clinical oncologist in practice in the community who is dedicating some of his time to assisting with the ongoing development of the ONCOCIN knowledge base. As an expert in oncology and in clinical trials, he takes primary responsibility for the quality and currency of the knowledge base. Christopher Lane (60%) is a systems programmer responsible for integrating and adapting the network virtual communications, shared task execution, and virtual graphics work with ONCOCIN core developments and dissemination experiments. He will also do the development of other ONCOCIN core system tools such as the object-oriented system. Mr. Samson Tu (50%), is a scientific programmer responsible for the EONYX research work under Dr. Fagan's direction. Mr. David Combs (50%), is a scientific programmer responsible for the EOPAL and METAOPAL research described in the body of the proposal. Ms. Janice Rohn (100%) is the data manager and oversees the clinic operation on a day-to-day basis. She also assists in data collection analysis for evaluation of ONCOCIN. Ms. Alison Grant (50%) is secretary for the ONCOCIN Project and co-ordinates all day-to-day office activities.

System Operations Support

Mr. Tucker (20%) is the Operations Manager and is responsible as our network liaison and for technical aspects of software export and overseeing system operations and backup. Ms. Ryalls (20%) acts as the system administrator, taking care of file space and directory allocations, providing system and user support for the resource, accounting, and assisting new projects get started on the resource.

Consultant

We do not plan any consulting support this renewal term.

Equipment Purchase

$14,000 per year is allocated for minor equipment purchases for the resource including communications equipment, Ethernet interfaces, local network gateway and TIP equipment, and workstation memory. We also allocate $75,000 per year for experimental Lisp workstations to support our core system development and dissemination studies. During the first year we expect to buy 4 Xerox 6045-based machines which will market for $18,000-19,000 each. In future years we will select from available machines such as the Texas Instruments VLSI-based machine that is being developed under DARPA funding, a machine that Hewlett Packard is developing, and announcements expected from Japanese manufacturers. These machines will allow us to remain current with the rapidly developing Lisp machine market for our own system development and also to maximize the service we can provide to the national community in developing applicable software for systems that those groups may purchase. This budget is increased by 7% per year to accommodate inflation.

Supplies

Office supplies are budgeted at $4,350 based on our past experience. Computer supplies are budgeted at $4,250 projecting recent workstation operating experience and including paper, disks, tapes, labels, laser printer supplies and other supplies needed for the computer facility. Engineering supplies are budgeted at $7,500 to cover needed parts and spares for maintaining in-house equipment and developing, interfacing, and integrating new equipment. We plan for continued development of Ethernet services needed to support existing and new Lisp machines, printers, and file servers at SUMEX. We have budgeted a 7% per year increase for all supplies

Travel

The travel budget covers domestic travel for staff to professional meetings, management committee meetings, and AIM workshop meetings. We budget $9,500 total for 4 east coast trips ($1400 each), 2 midwest trips ($1,000 each), and 3 west coast trips ($633 each). Future years are inflated by 7% per year.

Other Expenses

Equipment and Software Maintenance

We budget $19,200 per year for community file server maintenance from DEC and third party vendors and $1,350 for Diablo printers and miscellaneous equipment. We budget $30,000 for Lisp machine maintenance. We have relatively little experience with these machines out of warranty but are basing this estimate on partial coverage of time and materials repairs. The contract maintenance prices for these workstations is so high per machine and multi-machine discounts are not available that T&M is a more cost-effective approach. The allocated amount provides for maintenance for 20 machines at an estimated $1,500 per machine per year average cost. We budget $1,800 for software lease costs for packages that are necessary and for which we cannot arrange free access. We have budgeted a 7% per year increase for maintenance costs.

Telephone Services

We budget $13,100 for staff office telephones, and $4,000 for dataphone services for local Stanford community dialup ports on the local network and home terminal telephones for staff and some core research personnel to maximize productive working hours (generally well in excess of 8 hours per day total). Again, these estimates are based on the current configuration of lines and average monthly charges. We periodically review these arrangements to maintain satisfactory service at minimum cost. We anticipate annual increases to average 7%.

Auxiliary Computer Services

We budget $3,000 to cover service charges for AIM community use of other Stanford campus computer resources that complement SUMEX facilities. These include partial use of the Stanford Computer Science Department Dover printer, core research use of the SCORE 2060 machine, and various services from the Stanford ITS facility. We have budgeted 7% increase for each subsequent year.

Services and Documentation

$1,000 is budgeted for current documentation on system facilities and machines and $2650 for technical books and publication expenses. $2,700 is budgeted for photo-reproduction and various technical services based on previous experience. Each following year will reflect a 7% increase.

Prorated 2060 Operations Costs

As mentioned earlier, we plan to phase out DRR support for the DEC 2060 mainframe resource over the 5-year term of this grant. We plan to do this gradually and responsibly so that our users can relocate to other facilities or move to workstation environments for their research. For the first year we allocate $347,582 to the resource budget, which is 80% of the estimated 2060 operating costs detailed in the following section.

1.3.2. 2060 Operations Budget

This section explains the details of the 2060 operations budget for the proposed five year grant term. The figures in this section represent the *total* estimated 2060 operating costs. Only prorated shares of these costs are allocated to the resource budget as we phase-out the 2060 from our operations in favor of the workstation technologies we will be developing. The phasing is linear over 5 years with 80% of the 2060 costs charged to the resource budget in renewal year 1 (grant year 14), 60% in year 2, 40% in year 3, 20% in year 4, and 0% in year 5. As before, indirect costs are not shown in the budget and will be awarded separately on the basis of Modified Total Direct Costs. The indirect cost rate of 69%, is based on an agreement with the Office of Naval Research (ONR) dated September 14, 1984.

Personnel

Mr. Rindfleisch (10%) and Mr. Yeager (10%) are responsible for overall 2060 facility implementation and management. Ms. McCabe (25%) provides facility administrative support.

The programming staff, Mr. Sweer (90%) and Mr. Gilmurray (30%) share joint responsibility for system development and maintenance, user assistance, subsystem and utility program development, and operational support. These duties include, for example, performance analysis and improvement, bug correction, bringing up new monitor releases, system communications support, special device drivers and diagnostics, scheduler changes to control system allocation, and system maintenance. They also share responsibility for the system software such as user utilities, languages, and network utilities.

Mr. Tucker (80%) is responsible for network vendor interfaces and overseeing system operations and backup. He is assisted in providing file system archive and retrieval service and backup dumps as well as system utility programming support by 3 students (currently Blattel, Dolhert, and Jong). Ms. Ryalls (80%) acts as the system administrator, providing both system and user support for the resource.

Mr. Nicholas Veizades (20%) and Mr. Israel Torres (20%) provide electronics support for system maintenance, including special purpose, in-house designed hardware and terminal and communications equipment.

Personnel estimates are again based on current salaries and are increased by 7% per year for inflation. Staff benefits rates are the same as calculated for the main resource budget.

Consultant

We do not plan any consulting support for the 2060 operations.

Equipment Purchase

We budget $6,000 for minor equipment purchases including communications equipment, Ethernet interfaces, accessories, and other equipment replacements. This budget is increased by 7% per year to accommodate inflation.

Supplies

Office supplies are budgeted at $920 based on past experience. Computer supplies are budgeted at $8,000 projecting recent operating experience and including paper, ribbons, disks, tapes, labels, and other supplies needed for the computer facility. Engineering supplies are budgeted at $1,500 to cover needed parts and spares for maintaining in-house equipment. We have budgeted a 7% per year increase for all supplies.

Travel

The travel budget covers domestic travel for staff to technical meetings. We budget 1 east coast trip at $1,500. Future years are inflated by 7% per year.

Other Expenses

Equipment and Software Maintenance

The 2060 hardware system is covered on a DEC maintenance contract costing $92,300 per year. We also budget $3,950 for DEC software maintenance to keep up with the latest releases and $6,800 for other software licenses, including NCPCALC, SPSS, and SCRIBE. We have budgeted a 7% per year increase for maintenance costs.

Services and Documentation

$1,200 is budgeted for providing users with up-to-date documentation on system facilities and subsystem programs. Substantial efforts continue to upgrade documentation for the user community. $625 is budgeted for technical books and publication services. $825 is budgeted for photo-reproduction and technical services. Each following year will reflect a 7% increase.

Telephone Services

We budget $2,935 for staff office telephones and $14,000 for dataphone services for local Stanford community dialup ports on the SUMEX Computer and home terminal telephones for staff to increase the hours they can work and facilitate their access to the system at off hours when problems arise. These estimates are based on the current configuration of lines and average monthly charges. We periodically review these arrangements to maintain satisfactory service at minimum cost. We anticipate annual increases to average 7%.

Network Communications Support

We budget $60,000 for continued TYMNET network services for remote SUMEX-AIM users. This amount is estimated based on projections from current experience for TYMNET services (including network interface lines, maintenance, and usage costs). In past years, these funds have been distributed directly from NIH/BRTP through the Rutgers University TYMNET contract so as to maximize the benefit of a volume discount. This may still prove to be the most cost-effective approach and we will work closely with NIH/BRTP to secure these important services at the lowest cost. We include a 7% per year inflation rate.

The SUMEX-AIM ARPANET connection costs are being borne by ARPA Information Processing Techniques Office in support of the Stanford Knowledge Systems Laboratory basic AI research contract. We expect this relationship to continue and that NIH will continue to benefit from this arrangement.

1.4. Biographical Sketches

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E.H. Shortliffe

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
Edward H. Shortliffe	Assoc. Prof. of Medicine	8/28/47

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE *(circle highest degree)*	YEAR CONFERRED	FIELD OF STUDY
Harvard College, Cambridge, MA	A.B.	1970	Applied Math & Comp.Sci
Stanford University School of Medicine, Stanford, CA	Ph.D.	1975	Med. Inf. Science
	M.D.	1976	Medicine

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, experience, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles and complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NOT EXCEED TWO PAGES.

7/76 - 6/77 Intern in Medicine, Massachusetts General Hospital, Boston, MA
7/77 - 6/79 Resident in Medicine, Stanford University Medical Center, Stanford
7/79 - 2/85 Assistant Professor of Medicine, Stanford University School of Medicine
10/79- 2/85 Assistant Professor of Computer Science (by courtesy), Stanford University
1/80 - 2/85 Co-principal Investigator and Medical Liaison, SUMEX-AIM Computing Resource, Stanford University, Stanford, CA
3/85 - Associate Professor of Medicine, Stanford University School of Medicine
3/85 - Associate Professor of Computer Science (by courtesy), Stanford University
3/85 - Principal Investigator, SUMEX-AIM Computing Resource

Honors:

Editorial Boards: Medical Decision Making; Computer Methods and Programs in Biomedicine; Lecture Notes in Medical Informatics; Research Notes in Artificial Intelligence.

Research Career Development Award, NLM, 1979-1984

Henry J. Kaiser Family Foundation Faculty Scholar in General Internal Med., 1983-1986

Selected Publications

Shortliffe, E.H. Computer-Based Medical Consultations: MYCIN, Elsevier/ North Holland, New York, 1976. Japanese language version by Bunkodo Blue Books, Tokyo, 1981 (translated by T. Kaminuma).

Buchanan, B.G. and Shortliffe, E.H. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading,MA: Addison-Wesley, 1984.

Clancey, W.J. and Shortliffe, E.H. Readings in Medical Artificial Intelligence: The First Decade. Reading, MA: Addison-Wesley, 1984.

Shortliffe, E.H., Axline, S.G., Buchanan, B.G., Merigan, T.C., and Cohen, S.N. "An artificial intelligence program to advise physicians regarding antimicrobial therapy." Comput. Biomed. Res. 6:544-560 (1973).

Shortliffe, E.H. and Buchanan, B.G. "A model of inexact reasoning in medicine." Math. Biosci. 23:351-379 (1975).

Shortliffe, E.H., Davis, R., Axline, S.G., Buchanan, B.G., Green, C.C., and Cohen, S.N. "Computer-based consultations in clinical therapeutics: explanation and rule-acquisition capabilities of the MYCIN system." Comput. Biomed. Res. 8:303-320 (1975).

Wraith, S.M., Aikins, J.S., Buchanan, B.G., Clancey, W.J., Davis, R., Fagan, L.M., Hannigan, J.F., Scott, A.C., Shortliffe, E.H., van Melle, W.J., Yu, V.L., Axline, S.G., and Cohen, S.N. "Computerized consultation system for selection of antimicrobial therapy." Amer. J. Hosp. Pharm. 33:1304-1308 (1976).

Biographical Sketch - Edward H. Shortliffe (con't)

Davis, R., Buchanan, B.G., and Shortliffe, E.H. "Production rules as an approach to knowledge-based consultation systems." Artificial Intelligence 8:15-45 (1977).

Scott, A.C., Clancey, W., Davis, R., and Shortliffe, E.H. "Explanation capabilities of knowledge-based production systems." Amer. J. Computational Linguistics, Microfiche 62, 1977.

Yu, V.L., Buchanan, B.G., Shortliffe, E.H., Wraith, S.M., Davis, R., Scott, A.C., Axline, S.G., and Cohen, S.N. "Evaluating the performance of a computer-based consultant." Comput. Prog. Biomed. 9:95-102 (1979).

Shortliffe, E.H., Buchanan, B.G., and Feigenbaum, E.A. "Knowledge engineering for medical decision making: A review of computer-based clinical decision aids." Proceedings of the IEEE, 67:1207-1224 (1979).

Shortliffe, E.H. "The computer as clinical consultant" (editorial). Arch. Int. Med. 140:313-314 (1980).

Fagan, L.M., Shortliffe, E.H., and Buchanan, B.G. "Computer-based medical decision making: from MYCIN to VM." Automedica 3:97-106 (1980).

Teach, R.L. and Shortliffe, E.H. "An analysis of physician attitudes regarding computer-based clinical consultation systems." Comput. Biomed. Res. 14:542-558 (1981).

Shortliffe, E.H. "The computer and clinical decision making: good advice is not enough" (guest editorial). IEEE Engineering in Medicine and Biology Magazine, 1(2):16-18 (1982).

Wallis, J. and Shortliffe, E.H. "Explanatory power for expert systems: studies in the representation of causal relationships for medical consultation systems." Meth. Info. Med., 21:127-136 (1982).

Gerring, P.E., Shortliffe, E.H., and van Melle, W. "The Interviewer/Reasoner model: an approach to improving system responsiveness in interactive AI systems." AI Magazine 3(4):24-27 (1982).

Suwa, M., Scott, A.C., and Shortliffe, E.H. "An approach to verifying completeness and consistency in a rule-based expert system." AI Magazine 3(4): 16-21 (1982).

Duda, R.O. and Shortliffe, E.H. "Expert systems research." Science, 220(4594):261-268 (1983).

Aikins, J.S., Kunz, J.C., Shortliffe, E.H., and Fallat, R.J. "PUFF: An expert system for interpretation of pulmonary function data." Comput. Biomed. Res., 16:199-208 (1983).

Langlotz, C.P. and Shortliffe, E.H. "Adapting a medical consultation system to critique physicians' therapy plans." Int. J. Man-Machine Studs., 19:479-496 (1983). Reprinted in Developments in Expert Systems (M.J. Coombs, ed.), pp. 77-94, London: Academic Press, 1984.

Kunz, J.C., Shortliffe, E.H., Buchanan, B.G., and Feigenbaum, E.A. "Computer-assisted decision making in medicine." Journal of Philosophy and Medicine 9:135-160 (1984).

Shortliffe, E.H. "Reasoning methods in medical consultation systems: artificial intelligence approaches." Comput. Prog. Biomed., 18:5-14 (1984).

Gordon, J. and Shortliffe, E.H. "A method for managing evidential reasoning in a hierarchical hypothesis space." Accepted for publication in Artificial Intelligence, Summer 1985.

Shortliffe, E.H. "Consultation systems for physicians: the role of artificial intelligence techniques." Proceedings of the Third National Conference, Canadian Society for Computational Studies of Intelligence, Victoria, British Columbia, 14 May 1980. Also in Readings in Artificial Intelligence, (B. Webber and N. Nilsson, eds.), Tioga Publishing Co., Menlo Park, CA, 1981.

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
FEIGENBAUM, Edward A.	Professor, Computer Science	January 20, 1936

EDUCATION (Begin with baccalaureate or other initial professional education and include postdoctoral training)

INSTITUTION AND LOCATION	DEGREE (circle highest degree)	YEAR CONFERRED	FIELD OF STUDY
Carnegie Institute of Technology, Pittsburgh, Pennsylvania	B.S.	1956	Electrical Enginee:
Carnegie Institute of Technology Pittsburgh, Pennsylvania	Ph.D.	1959	Industrial Admin.

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, exp ence, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles a complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO N EXCEED TWO PAGES.

EXPERIENCE

University of California, Berkeley

Associate Professor, School of Business Administration, 1964-1965
Assistant Professor, School of Business Administration, 1960-63
Research Appointment, Center for Human Learning, 1961-64
Research Appointment, Center for Research in Management Science, 1960-64

Stanford University, Stanford, California

Professor of Computer Science, 1969-
Principal Investigator, Heuristic Programming Project, 1965-
Principal Investigator, SUMEX-AIM Project, national computer resource for application of artificial intelligence to medicine and biology, 1978-1985
Chairman, Computer Science Department, 1976-1981
Associate Professor of Computer Science, 1965-68
Director, Stanford Computation Center, 1965-68
Professor (by Courtesy), Department of Psychology, 1976-

Member, Computer and Biomathematical Sciences Study Section, National Institutes of Health, Bethesda, Md., 1968-72.
Member, Committee on Mathematics in the Social Sciences, Social Science Research Council, New York, NY, 1977-78.
Member, Computer Science Advisory Committee, National Science Foundation, 1977-80
Member, Advisory Committee on Mathematics in Naval Research, NRC/ONR, 1979-82
Member, National Advisory Committee, University of Missouri Health Care Technology Center (previous)
Member, Editorial Board, Journal of Artificial Intelligence
Editor, Computer Science Series, McGraw-Hill Book Co., 1965-1979
President, American Association for Artificial Intelligence (AAAI), 1980-81
Member, Council of American Association for Artificial Intelligence (AAAI), 1979-82
Member, Council of Cognitive Science Society, 1979-82
Member, DARPA Advisory Committee

PROFESSIONAL SOCIETIES

American Association for Artificial Intelligence (President, 1980-81)
Cognitive Science Society (Member, Governing Board, 1979-)
American Association for the Advancement of Science, (Fellow, 1983-)
Association for Computing Machinery (Member of National Council of ACM, 1966-68)
American College of Medical Informatics (Fellow, 1984-)

CONSULTANTSHIPS

IntelliCorp
Sperry Corporation

BOOKS AND MONOGRAPHS

Feigenbaum, E.A., & McCorduck, P. (1983). *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World.* New York: Addison-Wesley

Barr, A., Cohen, P., & Feigenbaum, E.A. *eds.* (1981, 1982). *Handbook of Artificial Intelligence* (Three Volumes). Los Altos, CA: Wm. Kaufmann Inc.

Buchanan, B., Feigenbaum, E.A., Lindsay, R., & Lederberg, J. (1980) *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project.* New York: McGraw-Hill.

Feigenbaum, E.A., & Feldman, J. *eds.* (1963). *Computers and Thought.* New York: McGraw-Hill.

Feigenbaum, E.A., Newell, A., Tonge, F., Mealy, G., et al. (1961). *Information Processing Language V Manual.* Englewood Cliffs, N.J.: Prentice-Hall.

Feigenbaum, E.A. (1959). *An Information Processing Theory of Verbal Learning.* Santa Monica, The RAND Corporation Paper P-1817 (Monograph).

SOME RECENT AND SELECTED PAPERS

Feigenbaum, E.A., & Simon, H. (1984) EPAM-like Models of Recognition and Learning *Cognitive Science 8,* 4 (Oct.-Dec.), 305-336.

Kunz, J., Feigenbaum, E.A., Buchanan, B., & Shortliffe, E.H. (1984). Comparison of Techniques of Computer-Assisted Decision Making in Medicine. in *Modeling and Analysis in Biomedicine.* Singapore: World Press. (335-367).

Kunz, J., Shortliffe, E.H., Buchanan, B.G., & Feigenbaum, E.A. (1984). Computer-Assisted Decision Making in Medicine. *Journal of Philosophy and Medicine, 9,* (135-160).

Feigenbaum, E.A. (1984). Knowledge Engineering: The Applied Side of Artificial Intelligence. In *Annals of the New York Academy of Sciences, 426,* (91-107).

Fagan, L.M., Kunz, J.C., Feigenbaum, E.A., & Osborn, J.J. (1979). Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence.* Tokyo, Japan. (216-262).

Fagan, L.M., Kunz, J.C., & Feigenbaum, E.A. (1979). A Symbolic Processing Approach to Measurement Interpretation in the Intensive Care Unit. In *Proceedings of the Third Annual Symposium on Computer Applications in Medical Care.* Silver Spring, Maryland.

Fagan, L.M., Kunz, J.C., Feigenbaum, E.A., & Osborn, J.J. (1979). Knowledge Engineering for Dynamic Clinical Settings: Giving Advice in the Intensive Care Unit. In *Proceedings of the Sixth IJCAI 79,* (260-262).

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
RINDFLEISCH, Thomas C.	Senior Research Associate/ Resource Director	December 10, 1941

EDUCATION (Begin with baccalaureate or other initial professional education and include postdoctoral training)

INSTITUTION AND LOCATION	DEGREE (circle highest degree)	YEAR CONFERRED	FIELD OF STUDY
Purdue University, Lafayette, Indiana	B.S.	1962	Physics
California Institute of Technology, Pasadena	M.S.	1965	Physics
	Ph.D.	Thesis to be completed; all cour work and examinations completed.	

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, exper ence, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles ar complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NO EXCEED TWO PAGES.

Stanford University:

1985 - present	Senior Research Associate/Director, Knowledge Systems Laboratory (KSL), Department of Computer Science, and Director, Symbolic Systems Resources Group (SSRG), including SUMEX, Department of Medicine
1982 - 1985	Senior Research Associate/Director, Heuristic Programming Project (HPP), Department of Computer Science
1976 - 1982	Senior Research Associate/Director, SUMEX Computer Resource, Departments of Medicine and Computer Science
1974 - 1976	Research Associate/Director, SUMEX Computer Project, Departments of Medicine and Computer Science
1971 - 1976	Research Associate, DENDRAL Project, Department of Genetics

Jet Propulsion Laboratory, California Institute of Technology, Pasadena:

1969 - 1971	Supervisor of Image Processing Development and Applications Group
1968 - 1969	Mariner Mars 1969 Cognizant Engineer for Image Processing
1962 - 1968	Engineer, design and implement image processing computer software

PUBLICATIONS (see attached sheet)

Publications:

1. Rindfleisch, T. and Willingham, D., "A Figure of Merit Measuring Picture Resolution," JPL Technical report 32-666, September 1965.
2. Rindfleisch, T. and Willingham, D., "A Figure of Merit Measuring Picture Resolution," Advances in Electronics and Electron Physics, Volume 22A, Photo-Electronic Image Devices, Academic Press, 1966.
3. Rindfleisch, T., "A Photometric Method for Deriving Lunar Topographic Information," JPL Technical Report 32-786, September 1965.
4. Rindfleisch, T., "Photometric Method for Lunar Topography," Photogrammetric Engineering, March 1966.
5. Rindfleisch, T., "Generalizations and Limitations of Photoclinometry," JPL Space Science Summary, Volume III, 1967.
6. Rindfleisch, T., "The Digital Removal of Noise from Imagery," JPL Space Science Summary 37-62, Volume III, 1970.
7. Rindfleisch, T., "Digital Image Processing for the Rectification of Television Camera Distortions," Astronomical Use of Television Type Image Sensors, NASA Special Publication SP-256, 1971.
8. Rindfleisch, T., Dunne, J., Frieden, H., Stromberg, W., and Ruiz, R., "Digital Processing of the Mariner 6 and 7 Pictures," Journal of Geophysical Research, Volume 76, Number 2, January 1971.
9. Pereira, W. E., Summons, R. E., Reynolds, W. E., Rindfleisch, T. C. and Duffield, A. M., "The Quantitation of Beta-Aminoisobutyric Acid in Urine by Mass Fragmentography," Clinica Chimica Acta, 49, 1973.
10. Summons, R. E., Pereira, W. E., Reynolds, W. E., Rindfleisch, T. C., and Duffield, A. M., "Analysis of Twelve Amino Acids in Biological Fluids by Mass Fragmentography," Analytical Chemistry, Vol. 46, No. 4, April 1974.
11. Pereira, W. E., Summons, R. E., Rindfleisch, T. C., and Duffield, A. M., "The Determination of Ethanol in Blood and Urine by Mass Fragmentography," Clin. Chim. Acta, 51, 1974.
12. Pereira, W. E., Summons, R. E., Rindfleisch, T. C., Duffield, A. M., Zeitman, B., and Lawless, J. G., "Stable Isotope Mass Fragmentography: Quantitation and Hydrogen-Deuterium Exchange Studies of Eight Murchison Meteorite Amino Acids," Geochem. et Cosmochim. Acta, 39, 163, 1975.
13. Dromey, R. G., Stefik, M. J., Rindfleisch, T. C., and Duffield, A. M., "Extraction of Mass Spectra Free of Background and Neighboring Component Contributions from Gas Chromatography/Mass Spectrometry Data", Analytical Chemistry, 48, page 1368, 1976.
14. Smith, D. H., Yeager, W. J., Anderson, P. J., Fitch, W. L., Rindfleisch, T. C., and Achenbach, M., "Historical Library Search. An Approach to Quantitative Comparison of GC/MS Profiles of Complex Mixtures," Analytical Chemistry, 49, page 1623, 1977.
15. Rindfleisch, T. C., Smith, D. H., Yeager, W. J., Achenbach, M, W., and Wegmann, A., "Advances in Data Acquisition and Analysis Systems for Applications of Gas Chromatography/Mass Spectrometry," in Biomedical Applications of Mass Spectrometry (First Supplementary Volume), edited by G. R. Waller and O. C. Dermer, page 55, John Wiley & Sons, New York, 1980.
16. Feigenbaum, E. A., Brown, H., Delagi, B. A., Gabriel, R. P., Nii, H. P., and Rindfleisch, T. C., "Advanced Architectures Project: Scope and Approach," Stanford Heuristic Programming Project Report HPP-84-40, October 1984.

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
YEAGER, William J.	Systems Programmer/Assistant Director	June 16, 1940

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE *(circle highest degree)*	YEAR CONFERRED	FIELD OF STUDY
University of California, Berkeley	B.A.	1964	Mathematics
California State University, San Jose	M.A.	1967	Mathematics
University of Washington, Seattle Doctoral studies (1969-70)	None	--	Mathematics

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, experience, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles and complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NOT EXCEED TWO PAGES.

1985 - present	Assistant Director, SUMEX Computer Project, Department of Medicine, Stanford University
1978 - 1985	**Systems Programmer,** SUMEX Computer Project, Department of Medicine, Stanford University
1975 - 1978	Scientific Programmer, Instrumentation Research Laboratories, Department of Genetics, Stanford University
1971 - 1975	Programmer, Bendix Field Engineering, Moffett Field, California
1970 - 1971	Programmer, WELLSCO Data Corp., San Francisco, California
1968 - 1969	Mathematics Instructor, Gavilan Jr. College, Gilroy, California
1967 - 1968	Mathematics Instructor, California Western Univ., San Diego
1966 - 1967	Mathematician/Programmer, Applied Physics Laboratory, Seattle, Washington
1966	Systems Representative, Burroughs Corp., San Jose, California

PUBLICATIONS Technical Report (Pending): Yeager, W.J.: "Ether TIPs and Gateways at SUMEX."

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
JACOBS, Charlotte	Asst Prof of Medicine	January 27, 1946

EDUCATION (Begin with baccalaureate or other initial professional education and include postdoctoral training)

INSTITUTION AND LOCATION	DEGREE (circle highest degree)	YEAR CONFERRED	FIELD OF STUDY
University of Rochester, Rochester, NY	B.A.	1968	Biology
Washington University School of Medicine, St. Louis, MO	M.D.	1972	
	Int,Jr Res,	1972 - 1974	Medicine
Univ of Ca, San Francisco, San Francisco,CA	Sr Res	1974 - 1975	Medicine
Stanford Univ, Stanford, CA 94305	Fellow	1975 - 1977	Oncology

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, experience, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles and complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NOT EXCEED TWO PAGES.

POSITIONS

1977 - 1980 Acting Assistant Professor, Department of Medicine, Division of Medical Oncology, Stanford University Medical Center, Stanford, CA

1977 - Present Director, Oncology Day Care Center, Department of Medicine, Stanford University Medical Center, Stanford, CA

1980 - Present Assistant Professor, Department of Medicine, Division of Medical Oncology, Stanford University Medical Center, Stanford, CA

OTHER EXPERIENCE

Drug Advisory Board, FDA (1984 - 1986)
Head and Neck Intergroup, Chairman (1984 - 1986)
Faculty Senate (1984 - 1986)

HONORS

Phi Beta Kappa
Alpha Omega Alpha
Kaiser Award for Excellence in Teaching (1983, 1985)
American Cancer Society Junior Faculty Clinical Fellowship (1981)
Janet Glasgow Scholastic Citation Award of the American Medical Women's Association (1972)
Missouri State Medical Association Award (1972)
Medical Alumni Scholarship Award (1971)
Lange Medical Book Awards (1969, 1970)
Janet Park Howell Award in Science (1968)

PUBLICATIONS

1. Jacobs C, Portlock CS, Rosenberg SA. Prednisone in MOPP chemotherapy for Hodgkin's disease. Br Med J 1976; 2:1469-1471.
2. Kim H, Jacobs C, Warnke RA, Dorfman RF. Malignant lymphoma with a high content of epitheloid histiocytes. Cancer 1978; 41:620-635.
3. Jacobs C, Bertino JR, Goffinet DR, Fee WE, Goode RL. Cis-platinum chemotherapy in head and neck cancers. Otolaryngol Head and Neck Surg 1978; 86:780-783.
4. Jacobs C, Bertino JR, Goffinet DR, Fee WE, Goode RL. 24-hour infusion of cis-platinum in head and neck cancers. Cancer 1978; 42:2135-2140.
5. Jacobs C. Hodgkin's disease - a patient teaching tool. Cancer Nursing 1979; 86:780-783.
6. Jacobs C. The role of cisplatin in the treatment of recurrent head and neck cancer. Cisplatin Current Status and New Developments. Edited by Prestayko AW, Crooke ST, Carter SK. Academic Press, 1980; 423-430.

DO NOT TYPE IN THIS SPACE—BINDING MARGIN

7. Levi J, Jacobs C, Kalman SM, McTigue M, Weiner MW. Mechanism of cis-platinum nephrotoxicity: I. Effects on sulfhydryl groups in rat kidneys. J Pharmacol Exp Ther 1980; 213:545-550.
8. Dobyan DC, Levi J, Jacobs C, Kosek J, Weiner MW. Mechanism of cis-platinum nephrotoxicity: II. Morphologic observations. J Pharmacol Exp Therap 1980; 213:551-556.
9. Jacobs C, Kalman SM, Tretton M, Weiner MW. Renal handling of cis-diamminedichloroplatinum (II) Cancer Treat Rep 1980; 64:1223-1226.
10. Jacobs C. High-dose methotrexate and cis-platinum in the treatment of recurrent head and neck cancer. Recent Results Cancer Res 1981; 76:290-295.
11. Jacobs C, Donaldson SS, Rosenberg SA, Kaplan HS. Management of the pregnant patient with Hodgkin's disease. Ann Intern Med 1981; 95:669-675.
12. Jacobs C, Ross R. The psychological assessment of cancer patients. Recent Advances in Clinical Oncology. Edited by Williams CJ, Whitehouse JMA. Churchill Livingstone, 1982; 365-374.
13. Mead G, Jacobs C. The changing role of chemotherapy in the management of head and neck cancer. Am J Med 1982; 73:582-595.
14. Jacobs C. Chemotherapy and combined modality treatment of head and neck cancer. Current Concepts in Oncology, Vol 4, No. 3, 1982.
15. Jacobs C. The use of methotrexate + 5-fluorouracil for recurrent head and neck cancer. Cancer Treat Rep 1982; 66:1925-1928.
16. Jacobs C, Ross R, Walker I, Stockdale FE. Behavior of cancer patients: A randomized study of the effects of education and peer support groups. Am J Clin Oncol 1983; 6:347-350.
17. Jacobs C, Meyers F, Hendrickson C, Kohler M, Carter S. A randomized phase III study of cisplatin with or without methotrexate for recurrent squamous cell carcinoma of the head and neck. Cancer 1983; 52:1563-1569.
18. Weiner MW, Jacobs C. Mechanism of cisplatin nephrotoxicity. Fed Proc 1983; 42:2974-2978.
19. Campbell AB, Kalman S, Jacobs C. Plasma platinum levels: Relationship to cisplatin dose and nephrotoxicity. Cancer Treat Rep 1983; 67 (2):169-172.
20. Coleman CN, Friedman MK, Jacobs C et al. Phase I trial of intravenous Melphalan plus the sensitizer Misonidazole. Cancer Res 1983; 43:5022-5025.
21. Jacobs C. The use of chemotherapy in the combination with radiotherapy in the treatment of head and neck squamous cancers. Advances in Treatment and Research. Edited by Wolf GT. Martinus Nijhoff Publishers, Boston, MA, 1984:265-286.
22. Jacobs C, Coleman CN, Rich L, Hirst K, Weiner MW. Inhibition of cisplatin secretion by the human kidney with probenecid. Cancer Res 1984; 44:3632-3635.
23. Jacobs C. The biophysiology of antineoplastic chemotherapy for head and neck cancers. Otolaryngology/Head and Neck Surgery. Edited by Cummings, Frederickson, Harker, Krause, Schuller. C.V. Mosby Company, St. Louis, MO 1985 (In press).
24. Shortliffe EH, Scott AC, Bischoff MD, Campbell AB, van Melle W, Jacobs C. An expert system for oncology protocol management. Rule-Based Expert Systems. The Mycin Experiments of the Stanford Heuristic Programming Project. Edited by Buchanan BG, Shortliffe EH. Addison-Wesley Company, Menlo Park, CA 1984:653-655.
25. Schreiber D, Jacobs C, Rosenberg SA, Cox R, Hoppe RT. The potential benefits of therapeutic splenectomy in Hodgkin's disease and non-Hodgkin's lymphomas. Int J Oncol Biol Phys 1984; 11:31-36.
26. Jacobs C, Hoppe RT. Non-Hodgkin's lymphomas of the head and neck extranodal sites. Int J Radiat Oncol Biol Phys 1984; 11:357-364.
27. Jacobs C. The role of chemotherapy in the treatment of head and neck cancer. Cisplatin Current Status and New Developments. Academic Press, 1985 (In Press).
28. Connors JM, Andiman WA, Howarth CB, Liu E, Merigan TC, Savage ME, Jacobs C. Treatment of nasopharyngeal carcinoma with human leukocyte interferon. J Clin Oncol 1985 (In Press).

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
Bruce G. Buchanan	Research Professor of Computer Science/	7-7-40

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE (circle highest degree)	YEAR CONFERRED	FIELD OF STUDY
Ohio Wesleyan University	B.A.	1961	Mathematics
Michigan State University	M.A.	1966	Philosophy
Michigan State University	Ph.D.	1966	Philosophy

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, experience, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles and complete references to all publications during the past three years and to representative earlier publications pertinent to this application. **DO NOT EXCEED TWO PAGES.**

Teaching and Professional Appointments

1981-present	Professor of Computer Science (Research) Stanford University Co-Principal Investigator (with E. Feigenbaum) of the Heuristic Programming Project since 1976.
1976-1981	Adjunct Professor, Computer Science Dept., Stanford
1972-1976	Research Computer Scientist, Computer Science Department, Stanford University
1966-1971	Research Associate, Artificial Intelligence Project, Stanford University
1965-1966	Instructor, Department of Philosophy Michigan State University

Professional Activities

- Editorial Board, Artificial Intelligence: An International Journal.
- Editorial Board, Journal of Automated Reasoning.
- Editorial Board, MIT Press series on Artificial Intelligence.
- Editorial Board, Addison-Wesley Press series on Expert Systems.
- Advisory Board, IEEE Transactions on Pattern Analysis and Machine Intelligence
- American Association for Artificial Intelligence -- Founding Committee, Program Committees, and Membership Chairman
- National Research Council Panel on Basic and Applied Research in Computer Science (1982-83)
- Teknowledge Inc. -- Co-Founder, Past President, Consulting Senior Scientist, Technology Advisory Board Member
- Comtex Scientific Corp. -- Scientific Advisory Board

Continued

RECENT AND SELECTED PUBLICATIONS

E.H. Shortliffe and B.G. Buchanan, "A Model of Inexact Reasoning in Medicine," Mathematical Biosciences, 23, 351, 1975.

B.G. Buchanan and E.A. Feigenbaum, "DENDRAL and Meta DENDRAL: Their Applications Dimension," Artificial Intelligence, 11 (1.2), 5, 1978.

E.H. Shortliffe, B.G. Buchanan, and E.A. Feigenbaum, "Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids," Proceedings of the IEEE, 67, 1207-1224, 1979.

R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum, and J. Lederberg, Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project, New York: McGraw-Hill, 1980.

B.G. Buchanan, "Research on Expert Systems," in J.E. Hayes, D. Michie, and Y.H. Pao (eds.), Machine Intelligence 10, New York: John Wiley, 1982.

B.G. Buchanan, "Partial Bibliography of Work on Expert Systems," SIGART Newsletter No. 84, (Association for Computing Machinery), April, 1983

Thomas G. Dietterich and B.G. Buchanan, "The Role of Experimentation in Theory Formation," In Proceedings of the International Workshop on Machine Learning June 1983, Univ. of Illinois at Urbana-Champaign, pp. 147-155.

B.G. Buchanan, "Introduction to the Memo Series of the Stanford Artificial Intelligence Laboratory," AI MAGAZINE Vol. 4, No. 4, Winter 1983.

B.G. Buchanan and E.H. Shortliffe, RULE-BASED EXPERT SYSTEMS: THE MYCIN EXPERIMENTS OF THE STANFORD HEURISTIC PROGRAMMING PROJECT. New York: Addison-Wesley, 1984.

J.C. Kunz, E.H. Shortliffe, B.G. Buchanan, and E.A. Feigenbaum, "Computer-Assisted Decision Making in Medicine," Journal of Medicine and Philosophy 9:135-160, 1984.

B.G. Buchanan, "Expert Systems," Journal of Automated Reasoning Vol. 1, No. 1, Winter 1985.

D.C. Wilkins, B.G. Buchanan, and W.J. Clancey, "Inferring an Expert's Reasoning by Watching." Proceedings of the 1984 Conference on Intelligent Systems and Machines, 1984. (Also HPP Report 84-29)

B.G. Buchanan, "Expert Systems: Toward Machines that Think." 1985 Yearbook of Science and the Future. Chicago: Encyclopaedia Britannica, Inc., 1984.

Li-min Fu and B.G. Buchanan, "Enhancing Performance of Expert Systems by Automated Discovery of Meta-Rules," Proceedings of IEEE Conference on Applications of Expert Systems 1984. (Also HPP Report HPP 84-38)

Li-Min Fu and B.G. Buchanan, "Learning Intermediate Knowledge in Constructing a Hierarchical Knowledge Base," to be presented at IJCAI-1985 and to appear in the conference proceedings.

R.G. Smith, H.A. Winston, T.M. Mitchell, and B.G. Buchanan, "Representation and Use of Explicit Justifications for Knowledge Base Refiniement," to be presented at IJCAI-85 and to appear in the conference proceedings.

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: Edward H. Shortliffe

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
Lawrence M. Fagan	Senior Research Associate	1/22/51

EDUCATION (Begin with baccalaureate or other initial professional education and include postdoctoral training)

INSTITUTION AND LOCATION	DEGREE (circle highest degree)	YEAR CONFERRED	FIELD OF STUDY
Massachusetts Institute of Technology	B.S.	1973	Interdisciplinary Sci
Stanford University, Stanford, California	Ph.D.	1980	Computer Science
University of Miami, Miami, Florida	M.D.	1983	Medicine

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, experience, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles and complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NOT EXCEED TWO PAGES.

Professional Experience

1972 - 1973 Research Assistant, Architecture Machine Group, MIT, Cambridge, Mass.

1973 - 1975 Research Associate; Systems Analyst, Program Verification Project, University of Southern California, Information Sciences Institute, Marina del Rey, Calif.

1975 Teaching Assistant, Department of Computer Science, Stanford University

1975 - 1979 Research Assistant, Heuristic Programming Project, Department of Computer Science, Stanford University, MYCIN Project: Diagnosis and therapy of infectious diseases. PUFF/VM: Expert System for intensive care units.

1979-1980 Associate Scientist, The Institues of Medical Sciences, Pacific Medical Center, San Francisco, Calif. (PUFF/VM Project)

1980 Research Associate, Heuristic Programming Project, Department of Computer Science, Stanford University: ONCOCIN Project.

1980 - 1981 Research Associate, Joint Appointment, Department of Medicine and Computer Science, Stanford University, Stanford, Calif. ONCOCIN Project.

1981 - 1983 On leave: Ph.D to M.D. Program, University of Miami, Miami, Florida.

July 1983 Senior Research Associate in Medicine, Department of Medicine, Stanford University.

Awards and Offices

1979 Paper of the Year - Medical Instrumentation

1980 Chairman, Program Demonstration Sessions, Artificial Intelligence in Medicine Conference, Stanford University, Stanford, California

1980 - Member, Executive Committee, Heuristic Programming Project

1983 Co-director, Medical Information Sciences Program, Stanford University

1983 National Liaison - SUMEX-AIM Computer National Resource for Artificial Intelligence in Medicine

1983 - Project Director: ONCOCIN, Cancer protocol management system.

Publications

1. Wraith, S.M., Aikins, J.S., Buchanan, B.G., Clancey, W.J., Davis, R., Fagan, L.M.et al. Computerized consultation system for selection of antimicrobial therapy, Americal Journal Hospital Pharmacy 33:1304-1308, 1976.
2. Yu, V.L., Fagan, L.M., Wraith, S.M.,et al. Antimicrobial selection by a computer - A blinded evaluation by infectious disease experts. J. Amer. Med. Assoc. 242:1279-1282, 1979.
3. Osborn, J.J., Fagan, L.M., Fallat, R., Kunz, J.C., McClung, D., Mitchell, R. Managing the data from respiratory measurements. Med. Instrumentation, 13(6), November 1979.

Biographical sketch, Lawrence M. Fagan - continued

4. Fagan, L.M.: Representing time-dependent relations in a medical setting. Ph.D. Thesis, Computer Science Department, Stanford University, 1980. (Thesis advisor: Edward Feigenbaum).
5. Fagan, L.M.: Measurement interpretation in the intensive care unit. Fifth Illinois Conference on Medical Information Systems, Champaign, Illinois, May 1979, pp. 253-262.
6. Fagan, L.M., Kunz, J.C., Feigenbaum, E.A. and Osborn, J.J.: A symbolic processing approach to measurement interpretation in the intensive care unit. Proc. Third Annual Symposium Computer Applications in Medical Care, Silver Spring, Maryland, October, 1979, pp. 30-33.
7. Fagan, L.M., Kunz, J.C., Feigenbaum, E.A., Osborn, J.J.: Representation of dynamic clinical knowledge: Measurement interpretation in the intensive care unit., 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan, August 1979, pp. 260-262.
8. Fagan, L.M., Shortliffe, E.H. and Buchanan, B.G.: Computer-based medical decision making: From MYCIN to VM. Automedica 3(2), 1980, pp. 97-106. Also appears in "Readings in Medical Artificial Intelligence", (W. Clancey and E.H. Shortliffe, eds.) Addison-Wesley Publ. Co., 1984.
9. Kunz, J.C., Fallat, R.J., McClung, D.H., Osborn, J.J., Votteri, B.A., Nii, H.P., Aikins, J.S., Fagan, L.M., Feigenbaum, E.A.: A physiological rule based system for interpreting pulmonary function test results. Proc. Computers in Critical Care and Pulmonary Medicine, IEEE Press, 1979.
10. Fagan, L.M. Understanding Spoken Language. In "The Handbook of Artificial Intelligence" (A. Barr and E.A. Feigenbaum, eds.), Vol. 1, pp 323-362, Los Altos, California, 1981.
11. Shortliffe, E.H. and Fagan, L.M. Expert systems research: modeling the medical decision making process. In "An Integrated Approach to Monitoring" (J.S. Gravenstein, R.S. Newbower, A.K. Ream, and N.T. Smith, eds.), pp. 185-202, Woburn, MA. Butterworth's 1983.
12. Fagan, L.M., Kunz, J.C., Feigenbaum, E.A., Osborn, J.J. "Extensions to the Rule-Based Formalism for a Monitoring Task". In "Rule-Based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project" (B. Buchanan and E.H. Shortliffe, eds.) Addison-Wesley, 1984.
13. Shortliffe, E.H. and Fagan, L.M. Artificial intelligence: the expert systems approach to medical consultation. Proceedings of the 6th Annual International Symposium on Computers in Critical Care and Pulmonary Medicine, 4-7 June 1984, Heidelberg, Germany.
14. Horvitz, E.J., Heckerman, D.E., Nathwani, B.N., and Fagan, L.M. Diagnostic Strategies in the Hypothesis-Directed PATHFINDER System. HPP Memo 84-13. First Conference on Artificial Intelligence Applications, Dec. 5-7, 1984, Denver, Colorado.
15. Blum, R.L., Walker, M.G., and Fagan, L.M. Minimycin: A Miniature Rule-Based System. To appear in M.D. Computing.
16. Preston, K., Jr., Fagan, L.M., Huang, H.K., and Pryor, T.A. Computing in Medicine. To appear in "Computer", 1984. (Centennial Issue).

Invited Talks

National Computer Conference, Los Angeles, CA 1978
Artificial Intelligence in Medicine Workshop, 1979, 1980, 1983, 1984
IEEE Chapter Biomedical Engineering, San Francisco, CA, 1980
Computers in Medicine Conference, Stanford, CA, 1984
Physicians and Computers Conference, Las Vegas, Nevada 1984
Stanford University Obstetrics-Gynecology Grand Rounds, 1984

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
Barbara Hayes-Roth	Sr. Research Associate	1/14/49

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE *(circle highest degree)*	YEAR CONFERRED	FIELD OF STUDY
Boston University	A.B.	1971	Psychology
University of Michigan	M.S.	1973	Psychology
University of Michigan	Ph.D.	1974	Psychology

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, exper ence, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles an complete references to all publications during the past three years and to representative earlier publications pertinent to this application. **DO NO EXCEED TWO PAGES.**

Employment History:

1982-Present Senior Research Associate, Computer Science Department
Stanford University, Stanford, Ca.
1976-1982 Senior Psychologist/Computer Scientist, Information
Sciences Department, The Rand Corporation, Santa Monica, Ca.
1974-1976 Member of Technical Staff, Instructional Research Department
Bell Laboratories, Murray Hill, N.J.
1972-1974 Teaching Fellow, Department of Psychology
The University of Michigan, Ann Arbor, Mi.
1971 System Analyst, Deapartment of Marketing Research
The Gillette Company, Boston, Ma.
1969-1971 Consultant and Programming Instructor
Cambridge Computer Associates, Cambridge, Ma.

Consulting History:

1984-Present AI Research Center
FMC Corporation, Santa Clara, Ca.
1985 AI Research Group
Martin-Marietta, Denver, Co.
1984-1985 Knowledge Systems Division
Perceptronics, Menlo Park, Ca.
1982-1985 Cognitive Interface Department
Hewlitt-Packard, Palo Alto, Ca.
1979-1983 Cognitive Processes Panel
National Science Foundation, Washington, D.C.
1979-1980 Consulting Associate Professor
Psychology and Computer Science Departments, Stanford University, Stanford, Ca.
1978 Summer Study Group on Educational Testing, Falmouth, Ma.
National Institute of Education, Washington, D.C.
1977-1979 Adjunct Assistant Professor
University of California, Los Angeles

Teaching Experience:

1979-1982	Cognitive Processes in Planning, Stanford University
1977	Thinking, University of California, Los Angeles
1972-1974	Artificial Intelligence, Learning and Memory, Psychology as a a Natural Science, Psychology Teaching Practicum, Psychology Research Practicum
1969-1971	Computer Programming, Cambridge Computer Associates

Fellowships, Honors and Professional Memberships:

Phi Beta Kappa, 1971
National Institutes of Health Trainee, The University of Michigan, 1971-1972
National Science Foundation Fellow, The University of Michigan, 1972-1974
Nominated as Fellow to the Center for Advanced Studies in the Behavioral Sciences, Stanford University, 1980
Member, American Association for Artifical Intelligence
Member, Cognitive Science Society

Publications and Technical Reports:

Hayes-Roth, B. A blackboard architecture for control, Artificial Intelligence Journal, 1985, in press.
Hayes-Roth, B., and Hewett, M. Learning control heuristics in BB1. Stanford University, Stanford, Ca. Report HPP-85-2.
Hayes-Roth, B. BB1: An environment for building blackboard systems that control, explain, and learn about their own problem-solving behavior. Stanford University, Stanford, Ca. Report HPP-84-16.
Hayes-Roth, B. A blackboard model of control. Heuristic Programming Project, Stanford University, Stanford, Ca., Report HPP-83-38, August, 1983
Hayes-Roth, B. An Overview of the Blackboard Architecture. Heuristic Programming Project, Stanford University, Stanford, Ca., Report HPP-83-30, February, 1983
Thorndyke, P.W., and Hayes-Roth, B. Differences in spatial knowledge acquired from maps and navigation. Cognitive Psychology, 1982, 14, 560-589.
Hayes-Roth, B. A cognitive science approach to improving planning. Proceedings of the Cognitive Science Society, Berkeley, 1981.
Hayes-Roth, B. Opportunism in consumer behavior. Proceedings of the Association for Consumer Research, October, 1981.
Kanouse, D., and Hayes-Roth, B. Cognitive considerations in the design of product warnings. In Barofsky, I., Mazis, M., and Morris, L.A. (Eds.), Banbury Reports: Product Labeling and Health Risks, Cold Spring Harbor, N.Y., 1980.
Hayes-Roth,B., and Hayes-Roth, F. A cognitive model of planning. Cognitive Science, 1979, 3, 275-310.

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: E. H. Shortliffe

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
Harold Brown	Sr. Research Associate	

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE *(circle highest degree)*	YEAR CONFERRED	FIELD OF STUDY
University of Notre Dame	M.S.	1963	Mathematics
Ohio State University	Ph.D.	1966	Mathematics

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, exper ence, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles an complete references to all publications during the past three years and to representative earlier publications pertinent to this application. **DO NO EXCEED TWO PAGES.**

Experience:

1979 -	Senior Research Associate, Member Heuristic Programming Project, Department of Computer Science, Stanford University
1977 - 1979	Senior Project Scientist, NASA-Ames Research Center Institute for Advanced Computation, Sunnyvale, CA
1974 - 1977	Research Associate, Member Heuristic Programming Project, Department of Computer Science, Stanford University
1977	Lecturer, Information Science Department, University of California, Santa Cruz, CA
1971 - 1972 1973 - 1974	Visiting Professor, Department of Computer Science, Stanford University
1963 - 1975	Instructor, Assistant Professor, Assistant Chairman, Associate Professor, Professor, Mathematics Department, Ohio State University
Winter 1971, 1973, 1975	Visiting Professor, Mathematics, Rhine. Westf. Tech. Hoch., Aachen
1964 - 1970	Director, Associate Director, NSF - SSTP
1967 - 1968	Visiting Member, Courant Institute of Mathematical Sciences, New York University
1960 - 1963	Assistant to the Chairman, Mathematics Department, University of Notre Dame

Professional Memberships:

American Association for Artificial Intelligence
Association for Computing Machinery
Institute of Electrical and Electronic Engineers

Representative Publications:

Brown, H.D. and Stefik, M. The partitioning of concerns in digital system design. Proceedings of the MIT Conference on Advanced Research in ULSI, 1981.

Brown, H.D., Tong, C., and Foyster, G. Environment for circuit design. Computer, vol. 16, no. 12, December 1983.

Brown, H.D., Yan, J. and Foyster, G. An expert system for assigning mask levels. HPP Report 83-39, October 1983.

BIOGRAPHICAL SKETCH

Give the following information for key professional personnel listed on page 2, beginning with the Principal Investigator/Program Director. Photocopy this page for each person.

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
Hisako Penny Nii	Research Associate	

EDUCATION *(Begin with baccalaureate or other initial professional education and include postdoctoral training)*

INSTITUTION AND LOCATION	DEGREE *(circle highest degree)*	YEAR CONFERRED	FIELD OF STUDY
Stanford University	M.S.	1973	Computer Science
Tufts University	B.S.	1962	Mathematics

RESEARCH AND/OR PROFESSIONAL EXPERIENCE: Concluding with present position, list in chronological order previous employment, expe ence, and honors. Include present membership on any Federal Government Public Advisory Committee. List, in chronological order, the titles ar complete references to all publications during the past three years and to representative earlier publications pertinent to this application. DO NO **EXCEED TWO PAGES.**

Experience:

1976 - present Research Associate, Heuristic Programming Project, Stanford University, Stanford, Ca.

1973 - 1975 Associate Investigator for Computer Science, HASP Project Systems Control, Inc., Palo Alto, California.

1967 - 1968 Systems Engineering Advisor, International Business Machines, World Trade Asia Corporation, Tokyo, Japan.

1962 - 1967 Research Staff Programmer, International Business Machines Corporation, Thomas J. Watson Research Center, Yorktown, New York

1965 - 1967 Project Leader, Electronic Coding Pad (ECP) System, Designed and developed a multi-language, conversational programming and debugging system using display terminals on satellite computers.

1965 - 1966 Assistant Manager, Man-Computer Interaction Group. Designed and implemented an interactive System/360 Assembly Language interpreter. Supervised projects in computer-aided project management, interactive PL/1, and graphic modeling.

1963 - 1964 Programmer. World's Fair Lexical Processing System, translation from Russian to English using special-purpose computer.

1962 - 1963 Programmer, miscellaneous applications ranging from text processing to linear programming problems.

Recent Publications

Nii, H. Penny. "Research on Blackboard Architectures at the Heuristic Programming Porject", Heuristic Programming Project Memo HPP-85-24, May 1985. Also to appear in: "The Proceedings of the Workshop on AI and Distributed Problem Solving", 1985.

Feignebaum, E., H. Brown, B. Delagi, R. Gabriel, P. Nii, and T. Rindfleisch. "Advanced Architectures Project: Scope and Approach," Heuristic Programming Project Memo HPP-84-40, October, 1984.

Nii, H. Penny. "Signal-to-Symbol Transformation: Reasoning in the HASP/SIAP Program", Proc. of ICASSP'84.

Nii, H. Penny. "Signal-to-Symbol Transformation: A Summary of HASP/SIAP Case Study," Intellectual Leverage for the Information Society, Digest of Papers, Spring Compcon83, IEEE Catalog No. 83CH1856-4, pp. 120 - 125.

Nii, H. P., E. A. Feigenbaum, J. J. Anton, and A. J. Rockmore. "Signal-to-Symbol Transformation: HASP/SIAP Case Study", The Artificial Intelligence Magazine, Vol 3, No. 1, 1982.

Nii, H. P. An Introduction to Knowledge Engineering, Blackboard Model, and AGE, Heuristic Programming Project Memo HPP-80-29, Computer Science Dept., Stanford University, 1980.

Manuals

Aiello, N., H.P. Nii, and W.C. White. The Joy of AGE-ing: An Introduction of AGE-1 System, Heuristic Programming Project Memo HPP-81-23, Computer Science Dept., Stanford University, October 1981.

Aiello, N., H.P. Nii, and W.C. White. AGE System Reference Manual, Heuristic Programming Project Memo HPP-81-24, Computer Science Dept., Stanford University, October 1981.

Aiello, N. and Nii, H. P. BOWL: A Beginner's Program Using AGE, AGE Example Series: No. 1, Heuristic Programming Project Memo-81-26, 1980.

Aiello, N. and H.P. Nii. AGEPUFF: A Simple Event-Driven Program, AGE Example Series No. 2, Heuristic Programming Project Memo-81-25, 1981.

1.5. Current and Pending Support

The following lists the sources of current and pending support for key personnel in our proposal:

EDWARD H. SHORTLIFFE

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 10%

Agency: National Institutes of Health
ID Number: RR-01631
Project Title: Studies of the Dissemination of Consultation Systems
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $624,455
Period Covered: 7/1/83-6/30/86
Current Award: (7/1/84-6/30/85) $222,511
Percent Effort Committed To Project: 30%

Agency: National Institutes of Health
ID Number: LM-04136
Project Title: Therapy-Planning Strategies for Consultation by Computer
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $211,851
Period Covered: 8/1/83- 7/31/86
Current Award: (8/1/84-7/31/85) $69,875
Percent Effort Committed To Project: 5% (No salary)

Agency: National Science Foundation
ID Number: IST 83-12148
Project Title: Information Structure and Use in Knowledge-Based Expert Systems
Principal Investigators: Bruce G. Buchanan & Edward H. Shortliffe
Amount Awarded: $330,138 (Total Costs)
Period Covered: 3/1/84-2/28/87
Current Award: (3/1/85-2/28/86) $101,308 (Total Costs)
Percent Effort Committed To Project: 5%

Agency: National Institutes of Health
ID Number: 1 T32 LM07033
Project Title: Postdoctoral Training in Medical Information Science
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $903,718
Period Covered: 7/1/84-6/30/89
Current Award: (7/1/84-6/30/85) $79,059
Percent Effort Committed To Project: 15% (No salary)

Agency: Henry J. Kaiser Family Foundation
ID Number: None
Project Title: Henry J. Kaiser Faculty Scholar in General Internal Medicine
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $150,000
Period Covered: 7/1/83-6/30/86 (renewable until June 1988)
Current Award: (7/1/84-6/30/85) $50,000
Percent Effort Committed To Project: Supports time on other projects; provides 40% of salary plus unrestricted research funds

PENDING SUPPORT:

Agency: National Institutes of Health
ID Number: 1 RO1 LM04420-01
Project Title: Knowledge Management for Clinical Trial Advice Systems
Principal Investigator: Edward H. Shortliffe
Amount Requested: $314,707
Period Covered: 7/1/85-6/30/88
Percent Effort Committed To Project: 5%

Agency: National Center for Health Services Research / National Institutes of Health
ID Number: NCI/HS-05414
Project Title: Computer support for clinical research in the community
Principal Investigator: Edward H. Shortliffe
Amount Requested: $1,578,840
Period Covered: 10/1/85-9/30/89
Percent Effort Committed To Project: 10%

APPLICATIONS IN PREPARATION: NONE

EDWARD A. FEIGENBAUM

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 6%

Agency: Boeing Computing Services Company
ID Number: None
Project Title: Research on Blackboard Problem-Solving Systems
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $225,000 (Total Costs)
Period Covered: 2/1/85 - 1/31/86
Percent Effort Committed To Project: 5%

Agency: Defense Advanced Research Projects Agency
ID Number: N00039-83-C-0136
Project Title: Heuristic Programming Project
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $3,354,493 (Total Costs)
Period Covered: 10/1/82 - 9/30/85
Percent Effort Committed To Project: 10%

Agency: Defense Advanced Research Projects Agency
ID Number: MDA903-83-C-0188
Project Title: Research Computing Equipment Modernization
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $2,565,000 (Total Costs)
Period Covered: 6/1/83 - 5/31/85
Percent Effort Committed To Project: 0% salary

Agency: Defense Advanced Research Projects Agency
ID Number: F30602-85-C-0012
Project Title: Expert Systems on Multiprocessor Architecture
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $4,454,444 (Total Costs)
Period Covered: 3/14/85 - 3/13/89
Percent Effort Committed To Project: 19%

Agency: NASA-AMES Research Center
ID Number: NCC 2-220, S1
Project Title: Research on Advanced Knowledge-Based System Architectures
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $265,000 (Total Costs)
Period Covered: 10/1/82 - 11/30/85
Percent Effort Committed To Project: 2%

Agency: IBM; IBM/Stanford Joint Study
ID Number: None
Project Title: The Use of Design Models in the Diagnosis of Computer Hardware
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $846,824 (Total Costs)
Period Covered: 6/1/80 - 5/31/85
Percent Effort Committed To Project: 2%

Agency: National Science Foundation
ID Number: MCS-8310236
Project Title: Applications of AI to Molecular Biology
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $270,836 (Total Costs)
Period Covered: 11/1/83 - 10 /31/85
Current Award: (11/1/84 - 10/31/85) $131,621 (Total Costs)
Percent Effort Committed To Project: 4% (no current salary support)

Agency: National Science Foundation
ID Number: MCS-8303142
Project Title: The Mechanization of Formal Reasoning (Computer Research)
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $183,921 (Total Costs)
Period Covered: 7/15/83 - 6/30/85
Current Award: (7/1/84 - 6/30/85) $98,657 (Total Costs)
Percent Effort Committed To Project: 2% (no current salary support)

PENDING SUPPORT:

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed To Project: 15%

APPLICATIONS IN PREPARATION: NONE

Current and Pending Support

BRUCE G. BUCHANAN

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 10%

Agency: NASA-Ames Research Center
ID Number: NCC02-274
Project Title: Research On Knowledge Representation
Principal Investigator: Bruce G. Buchanan
Amount Awarded: $850,000 (Proposed Total Costs)
Period Covered: 10/1/83 - 9/30/88 (support level pending for future years)
Percent Effort Committed To Project: 10%

Agency: Defense Advanced Research Projects Agency
ID Number: N00039-83-C-0136
Project Title: Heuristic Programming Project
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $3,354,493 (Total Costs)
Period Covered: 10/1/82 - 9/30/85
Percent Effort Committed To Project: 40%

Agency: National Science Foundation
ID Number: IST-83-12148
Project Title: Information Structure-Use Knowledge-Based Expert Systems
Principal Investigators: Bruce Buchanan & Edward H. Shortliffe
Amount Awarded: $330,138 (Total Costs)
Period Covered: 3/15/84 - 2/28/87 (support level pending for future years)
Current Award: (3/1/85 - 2/28/86) $101,308
Percent Effort Committed To Project: 5%

Agency: National Science Foundation
ID Number: PCM-84-02348
Project Title: Interpretation of NMR Data for Proteins Using AI Methods
Principal Investigators: Bruce Buchanan & Oleg Jardetzky
Amount Awarded: $100,000 (Total Costs)
Period Covered: 11/1/84 - 10/31/86
Current Award: (11/1/84 - 10/31/85) $50,000 (Total Costs)
Percent Effort Committed To Project: 0% (No salaries included in grant.)

Agency: NASA-Goddard Space Flight Center
ID Number: NAG-5-261
Project Title: Planning in Uncertain and Unforgiving Situations, and Planning Physical Actions
Principal Investigators: Bruce G. Buchanan & Thomas O. Binford
Amount Awarded: $127,837 (Total Costs) (level pending for future years)
Period Covered: 9/1/83 - 8/31/85
Percent Effort Committed To Project: 12%

Agency: Josiah Macy, Jr. Foundation
ID Number: None
Project Title: A Family of Intelligent Tutoring Programs for Medical Diagnosis
Principal Investigator: Bruce G. Buchanan
Amount Awarded: $503,415 (Total Costs)
Period Covered: 3/1/85 - 2/29/88
Percent Effort Committed To Project: 5%

Agency: International Business Machines
ID Number: None
Project Title: Attempts to Determine the User's Conceptualization System
Principal Investigator: Bruce G. Buchanan
Amount Awarded: $165,000 (Total Costs)
Period Covered: 5/11/84 - 5/10/85
Percent Effort Committed To Project: 5%

Agency: Boeing Computing Services Company
ID Number: None
Project Title: Research on Blackboard Problem-Solving Systems
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $225,000 (Total Costs)
Period Covered: 2/1/85 - 1/31/86
Percent Effort Committed To Project: 10%

Agency: Lawrence Livermore
ID Number: None
Project Title: Research on Intelligent Budget Planning and Resource Management Systems
Principal Investigator: Bruce G. Buchanan
Amount Awarded: $49,964 (Total Costs)
Period Covered: 12/14/84 - 9/30/85
Percent Effort Committed To Project: 3%

PENDING SUPPORT:

Agency: National Institutes of Health
ID Number: None
Project Title: Understanding and Critiquing Clinical Trials Literature
Principal Investigators: Bruce G. Buchanan & B.W. Brown
Amount Requested: $340,316
Period Covered: 7/1/85 - 6/30/88
First Year: 7/1/85 - 6/30/86 $107,505
Percent Effort Committed To Project: 5%

Agency: Office of Naval Research
ID Number: None
Project Title: Computer-Based Tutors for Explaining and Managing the Process of Diagnostic Reasoning
Principal Investigator: Bruce G. Buchanan
Amount Requested: $510,622 (Total Costs)
Period Covered: 3/15/85 - 3/14/88
Percent Effort Committed To Project: 5%

Current and Pending Support

Agency: Office of Naval Research
ID Number: None
Project Title: Expert Control of Problem-Solving Search
Principal Investigator: Bruce G. Buchanan
Amount Requested: $725,899 (Total Costs)
Period Covered: 8/1/85 - 7/31/88
Percent Effort Committed To Project: 5%

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed To Project: 35%

APPLICATIONS IN PREPARATION: NONE

LAWRENCE M. FAGAN

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 5%

Agency: National Institutes of Health
ID Number: RR 01613
Project Title: Studies in the Dissemination of Consultation Systems
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $624,455
Period Covered: 7/1/83 - 6/30/86
Current Award: (7/1/84 - 6/30/85) $222,511
Percent Effort Committed To Project: 16%

Agency: National Institutes of Health
ID Number: LM-04136
Project Title: Therapy-Planning Strategies for Consultation by Computer
Principal Investigator: Edward H. Shortliffe
Amount Awarded: $211,851
Period Covered: 8/1/83-7/31/86
Current Award: (8/1/84-7/31/85) $69,875
Percent Effort Committed To Project: 29%

Agency: National Institutes of Health
ID Number: 1 R23 LM04316
Project Title: Explanation of Computer-Assisted Therapy Plans
Principal Investigator: Lawrence M. Fagan
Amount Awarded: $107,441
Period Covered: 2/1/85-1/31/88
Current Award: (2/1/85-1/31/86) $37,500
Percent Effort Committed To Project: 50%

PENDING SUPPORT: NONE

APPLICATIONS IN PREPARATION: NONE

Current and Pending Support

THOMAS C. RINDFLEISCH

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 100%

PENDING SUPPORT:

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed to Project: 20%

APPLICATIONS IN PREPARATION: NONE

WILLIAM J. YEAGER

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed To Project: 100%

PENDING SUPPORT: NONE

APPLICATIONS IN PREPARATION: NONE

Current and Pending Support

HAROLD BROWN

CURRENT SUPPORT:

Agency: Defense Advanced Research Projects Agency
ID Number: N00039-83-C-0136
Project Title: Heuristic Programming Project
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $3,354,493 (Total Costs)
Period Covered: 10/1/82 - 9/30/85
Percent Effort Committed To Project: 100%

Agency: Defense Advanced Research Projects Agency
ID Number: F30602-85-C-0012
Project Title: Expert Systems on Multiprocessor Architecture
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $4,454,444 (Total Costs)
Period Covered: 3/14/85 - 3/13/89
Percent Effort Committed To Project: 50%, beginning 10/85

PENDING SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-13
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Pending Award: (8/1/85 - 7/31/86) $1,281,295
Percent Effort Committed To Project: 10%

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed To Project: 10%

APPLICATIONS IN PREPARATION: NONE

HISAKO PENNY NII

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-12
Project Title: SU Medical Experiment Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $1,108,929
Percent Effort Committed to Project: 40%

Agency: Defense Advanced Research Projects Agency
ID Number: N00039-83-C-0136
Project Title: Heuristic Programming Project
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $3,354,493 (Total Costs)
Period Covered: 10/1/82 - 9/30/85
Percent Effort Committed To Project: 40%

Agency: Defense Advanced Research Projects Agency
ID Number: F30602-85-C-0012
Project Title: Expert Systems on Multiprocessor Architecture
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $4,454,444 (Total Costs)
Period Covered: 3/14/85 - 3/13/89
Percent Effort Committed To Project: 50%, beginning 8/85

Agency: NASA-AMES Research Center
ID Number: NCC 2-220, S1
Project Title: Research on Advanced Knowledge-Based System Architectures
Principal Investigator: Edward A. Feigenbaum
Amount Awarded: $265,000 (Total Costs)
Period Covered: 10/1/82 - 11/30/85
Percent Effort Committed To Project: 20%

PENDING SUPPORT:

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & B.G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed to Project: 5%

APPLICATIONS IN PREPARATION: NONE

Current and Pending Support

BARBARA HAYES-ROTH

CURRENT SUPPORT:

Agency: Defense Advanced Research Projects Agency
ID Number: N00039-83-C-0136
Project Title: Heuristic Programming Project
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Awarded: $3,354,493 (Total Costs)
Period Covered: 10/1/82 - 9/30/85
Percent Effort Committed To Project: 75%

PENDING SUPPORT:

Agency: National Institutes of Health
ID Number: 5 P41 RR00785-13
Project Title: SU Medical Experimental Computer Resource (SUMEX)
Principal Investigators: Edward H. Shortliffe & Edward A. Feigenbaum
Amount Awarded: $6,400,287
Period Covered: 8/1/81 - 7/31/86
Current Award: (8/1/85 - 7/31/86) $1,281,295
Percent Effort Committed To Project: 15%

Agency: Defense Advanced Research Projects Agency
ID Number: None
Project Title: Knowledge-Based Systems Research
Principal Investigators: Edward A. Feigenbaum & Bruce G. Buchanan
Amount Requested: $4,464,793 (Total Costs)
Period Covered: 10/1/85 - 9/30/88
Percent Effort Committed To Project: 50%

Agency: Office of Naval Research
ID Number: None
Project Title: Computer-Based Tutors for Explaining and Managing The Process of Diagnostic Reasoning
Principal Investigator: Bruce G. Buchanan
Amount Requested: $510,311 (Total Costs)
Period Covered: 3/15/85 - 3/14/88
Percent Effort Committed to Project: 25%

APPLICATIONS IN PREPARATION: NONE

CHARLOTTE JACOBS

CURRENT SUPPORT:

Agency: National Institutes of Health
ID Number: P01 CA34233-02
Project Title: Clinical and Laboratory Studies of the Malignant Lymphomas
Principal Investigator: Saul A. Rosenberg
Period Covered: 4/1/83 - 3/31/86
Current Award: (4/1/85 - 3/31/86) $1,314,907
Percent Effort Committed To Project: 10%

Agency: National Institutes of Health
ID Number: CA09287-07
Project Title: Investigative Oncology
Principal Investigator: Saul A. Rosenberg
Period Covered: 9/1/78 - 8/31/88
Current Award: (9/1/84 - 8/31/85) $92,580
Percent Effort Committed To Project: 3%

Agency: National Institutes of Health
ID Number: 1 R24-RR01631
Project Title: Studies in the Dissemination of Consultation Systems
Principal Investigator: Edward H. Shortliffe
Period Covered: 7/1/83 - 6/30/86
Current Award: (7/1/84 - 6/30/85) $222,511
Percent Effort Committed To Project: 5%

Agency: National Institutes of Health
ID Number: CA25862-04
Project Title: Northern California Oncology Group
Principal Investigator: C. Norman Coleman
Period Covered: 8/1/83 - 7/31/86
Current Award: (8/1/84 - 7/31/85) $104,483
Percent Effort Committed To Project: 3%

Agency: National Institutes of Health
ID Number: 1R01A2 CA33849
Project Title: Chemical Modifiers of Radiation Therapy and Chemotherapy
Principal Investigator: C. Norman Coleman
Period Covered: 5/1/84 - 4/30/87
Current Award: (5/1/85 - 4/30/86) $184,256
Percent Effort Committed To Project: 10%

PENDING SUPPORT:

Agency: National Institutes of Health
ID Number: 1R01CA3771-01
Project Title: Modifiers of Cisplatin Nephrotoxicity
Principal Investigator: Charlotte Jacobs
Amount Requested: $361,503
Period Covered: 7/1/85 - 6/30/88
Percent Effort Committed To Project: 30%

Current and Pending Support

Agency: Bristol Myers
ID Number: None
Project Title: Improvements in the Efficacy of Cisplatin Nephrotoxicity
Principal Investigator: Charlotte Jacobs
Amount Requested: $347,658
Period Covered: 7/1/85 - 6/30/86
Percent Effort Committed To Project: 30%

Agency: National Institutes of Health
ID Number: 2 R25 CA21555-07A1
Project Title: Cancer Education Program
Principal Investigator: Charlotte Jacobs
Amount Requested: $293,182
Period Covered: 7/1/85 - 6/30/90
Percent Effort Committed To Project: 25%

Agency: Bristol Myers
ID Number: None
Project Title: A Phase III Randomized Study Comparing High Dose Bolus Platinol (DDP) and 96-Hour Continuous Infusion Fluorouracil (5-FU) in Combination and as Single Agents in Advanced Squamous Cell Carcinoma of the Head and Neck
Principal Investigator: Charlotte Jacobs
Amount Requested: $21,600
Period Covered: 7/1/85 - 6/30/86
Percent Effort Committed To Project: 26%

APPLICATIONS IN PREPARATION: NONE

1.6. Resources and Environment

RESOURCES AND ENVIRONMENT

FACILITIES: Mark the facilities to be used at the applicant organization and briefly indicate their capacities, pertinent capabilities, relative proximit and extent of availability to the project. Use "other" to describe the facilities at any other performance sites listed in Item 9, page 1, and at sites fo field studies. Using continuation pages if necessary, include an explanation of any consortium arrangements with other organizations.

☐ Laboratory:

☐ Clinical:

☐ Animal:

☒ Computer: See Major Equipment paragraph below.

☐ Office:

☐ Other (____________________):

MAJOR EQUIPMENT: List the most important equipment items already available for this project, noting the location and pertinent capabilities each. SUMEX-AIM develops and operates a heterogeneous networked system of computing resources, including mainframe host computers, Lisp workstations, and network utility servers. Host machines include a DEC 2060 and 2020 running TOPS-20 and a VAX 11/780 running UNIX (these are the current core of the nationally available resource). Our Lisp workstations include more than 25 Xerox 11xx's, a Symbolics LM-2, eight Symbolics 36xx's, and five Hewlett-Packard 9836's. Network printing, file storage, gateway, and terminal interface services are provided by dedicated VAX 11/750's through an extensive Ethernet and to external resources through the ARPANET and TYMNET.

ADDITIONAL INFORMATION: Provide any other information describing the environment for the project. Identify support services such consultants, secretarial, machine shop, and electronics shop, and the extent to which they will be available to the project.

2. Resource Plan

Before launching into the technical details of our proposal, we want to explain two matters relating to its scope and organization:

Combined SUMEX-AIM and ONCOCIN renewal

This is an application for the 5-year merged renewal of two on-going Biomedical Research Technology Resource efforts: 1) the *S*tanford *U*niversity *M*edical *EX*perimental computer research resource for applications of *A*rtificial *I*ntelligence in *M*edicine (SUMEX-AIM, RR-00785) and 2) the resource-related research project for Studies in the Dissemination of Consultation Systems (ONCOCIN, RR-01631). We propose that the combined research activities of these projects be funded under a continuation of the SUMEX-AIM grant and that the core research aspects of the resource-related ONCOCIN work not be continued separately. The reasons for merging these renewals are both technical and administrative.

On the technical side, the goals for the two projects are inextricably mingled in the development and exploitation of AI techniques and Lisp workstation technology for experimental applications in medical decision-making systems. The recent ONCOCIN experiments in developing and disseminating a cancer chemotherapy protocol advisor (built on joint SUMEX/ONCOCIN system technology) have effectively demonstrated the viability of this applied technology. They have accordingly helped define important future directions for the longer term thrust of the SUMEX-AIM resource toward distributed workstations as the computing model for the next generation of biomedical AI systems.

On the administrative side, the current award periods for both grants end in mid-summer of 1986. Also, Professor Shortliffe is now Principal Investigator of both projects and there is no logical way to separate the management of such closely linked research efforts.

Length of this proposal

We have attempted to keep this proposal as brief as possible. However, we felt obliged to exceed some of the page limitations stipulated in the NIH guidelines for a several reasons.

First, the computer science discipline of artificial intelligence is relatively new and its intersection with and significance to medicine requires more explanation than more traditional areas of biomedical research. Second, the SUMEX-AIM resource encompasses a national community of more than 12 core research projects and 13 collaborative research projects pursuing diverse applications areas. In order to illustrate the scope of the community and provide the scientific basis for continued support of SUMEX as a resource, the objectives of these projects must be presented with enough detail to give reviewers unfamiliar with some aspects of the work a proper perspective. We also include a brief description of the important operational base of the resource. And finally, this application is for a 5-year renewal term. Many of the core and collaborative research efforts are aimed at long term goals to assist biomedical researchers and clinicians in information management, analysis, and decision making. In order to provide a more efficient research environment, avoiding the overhead of additional proposal preparations and reviews on time scales shorter than expected result horizons, we hope to describe our goals in sufficient detail to justify the 5-year award period.

2.1. Introduction and Background

2.1.1. Principal Investigators' Executive Summary

In the almost twelve years since the SUMEX-AIM resource was established, computing technology and biomedical artificial intelligence research have undergone a remarkable evolution. As we prepare this proposal to renew the resource through the remainder of the 1980's, we take pride in the realization that SUMEX has both influenced and responded to those changing technologies. It is widely recognized that our resource has fostered highly influential work in medical AI -- work from which it is generally acknowledged that the expert systems field emerged -- and that it has simultaneously helped define the technological base of applied AI research. The LISP machines 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. We look forward to another half decade during which the era of centralized machines for AI research will come to an end, having been supplanted by networks of distributed and heterogeneous single-user machines sharing common resources such as file servers, printers, and gateways to other local and long-distance networks.

Although we reflect on the past with pride and satisfaction, and continue to be motivated by the goals that led to the initiation of SUMEX over a decade ago, our present momentum and on-going accomplishments inevitably direct us to the future. We are delighted that our sense of excitement about this field and its evolution has been sustained and that the future holds both challenges and promise that continue to carry our research community forward. The "spirit of SUMEX" that was fostered by our past efforts and goals provides an on-going stimulus to innovation and accomplishment. However, the contributing parts of that spirit do not come across well in the dry recitations of a voluminous proposal document such as the one that follows. Thus we begin with this prelude that provides an overview of our accomplishments and our proposed future directions. As in the past, we continue to be motivated by three main goals:

1. to develop and provide impeccable computing resources and human assistance to scientists working on applications of artificial intelligence research in medicine and biology;

2. to demonstrate that it is feasible to provide resources and assistance to a national community of researchers from a central site, integrating distributed and centralized computing technology, local and national computer communication networks, and a staff oriented toward the special problems of individuals participating in AIM research at other institutions;

3. to develop the community of scientists interested in working on applications of AI to the biomedical sciences; facilitating the growth, health, and vigor of the community by providing electronic communications that link its members and by assisting with the dissemination of systems software and applications programs that are of use to the wider community of AIM researchers. One question we have been asking is, "Is there a new style of science that will emerge in a communications-enhanced setting of national, rather than institutional, scope?" Within a decade it was clear that the answer to this question was (and is) "yes"!

SUMEX's Success as a National Research Resource

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. Many NIH computer RR's have been mostly institutional in scope, occasionally regional (like the UCLA resource). SUMEX now has the reputation of 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.

A decade ago, when machines up to the task of supporting AI research cost $1M, some of the most notable projects in the history of Artificial Intelligence were done with terminal-and-network, without a computer on site. In human terms, this meant, of course, not having the headaches and energy drains of proposing a machine, installing it, maintaining it and its software, hiring its system programmers and operators, dealing with communication vendors, etc. The famous INTERNIST program was developed from Pittsburgh in this way. And the ACT computer model was begun at Michigan, continued at Yale, and later at Carnegie-Mellon, all without moving the program or losing a day's work because of machine transition problems. The GENET community of over 300 molecular biologists grew up in a year around SUMEX programs for analyzing DNA sequences. Their demand for these centralized capabilities ultimately swamped our machine and led to the initiation of a separate resource (BIONET) to meet their needs.

The projects SUMEX supports have generally required substantial computing resources with excellent interaction. Even today though, with the growing availability of Lisp workstations, this computing power is still hard to obtain in all but a few universities. SUMEX is, in a sense, a "great equalizer". A scientist gains access by virtue of the quality of his/her research ideas, not by the accident of where s/he happens to be situated. In other words, the resource follows the ethic of the scientific journal.

SUMEX has demonstrated that a computer resource is a useful "linking mechanism" for bringing together and holding together teams of experts from different disciplines who share a common problem focus. For example, computer scientists have been collaborating fruitfully with physical chemists, molecular biochemists, geneticists, crystallographers, internists, ophthalmologists, infectious disease specialists, intensive care specialists, oncologists, psychologists, biomedical engineers, and other expert practitioners. And in some of these cases, the interdisciplinary collaboration, usually so difficult to achieve in the best of circumstances, was achieved in spite of geographical distance between the participants, using the computer networks.

SUMEX has also achieved successes as a community builder. 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. Over 36 biomedical AI application projects have developed in our national community and have been supported by SUMEX over the years. And 9 of these have matured to the point of now continuing their research on facilities outside of SUMEX. For example, the BIONET resource (named GENET while at SUMEX) is being operated by IntelliCorp; the Rutgers Computers in Biomedicine resource is centered at Rutgers University; the CADUCEUS project splits their research work between their own VAX computer and the SUMEX resource; and the Chemical Synthesis project now operates entirely on a VAX at U.C. Santa Cruz.

The SUMEX mission has been able to capture the contributions of some of the finest computers-in-medicine specialists and computer scientists in the country. For example, Professor Joshua Lederberg (SUMEX's first PI, now President of The Rockefeller University) is a member of SUMEX's Executive Committee; and Dr. Donald Lindberg, former Director of the University of Missouri's Medical Information Science group, and now Head of the National Library of Medicine, was until recently the Chairman of the AIM Advisory Group. Professor Herbert Simon of Carnegie-Mellon University, Professor Marvin Minsky of MIT, and many other distinguished scientists serve on that peer review committee.

SUMEX and Artificial Intelligence Research

The SUMEX Project is a relative latecomer to AI research. Yet its scope has given strong impetus to this historic development in applied computer science. AI research is that part of computer science that investigates symbolic reasoning processes, and the representation of symbolic knowledge for use in inference. It views heuristic or judgmental knowledge to be of equal importance with "factual" knowledge, indeed to be the essence of what we call "expertise". In its "Expert Systems" work, it seeks to capture the expertise of a field, and translate it into programs that will offer intelligent assistance to a practitioner in that field.

For computer applications in medicine and biology, this research path is crucial, indeed ineluctable. 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.

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. The AI methodology is beginning to fill that need. Heuristic knowledge can be elicited, studied, critiqued by peers, and taught to students.

The tide of AI research and application is rising. AI is one of the principal fronts along which university computer science groups are expanding. Federal and industrial support for AI research is vigorous and growing, although support specifically for biomedical applications continues to be limited. The pressure from student career-line choices is great: to cite an admittedly special case, approximately 80% of the students applying to Stanford's computer science Ph.D. program cite AI as a possible field of specialization (up from 30% 4 years ago). At Stanford, we have vigorous special programs for student training and research in AI -- a new 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 our 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.

There is 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, over 1000 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 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.

In industry, AI is on an exponential growth path as well. In the USA alone, over 30 AI start-up companies have been formed in the past four years and many groups have been established in large companies as well. The list of names is long and includes Hewlett-Packard, Schlumberger (including Fairchild), Texas Instruments, Xerox, IBM, DEC, General Motors, General Electric, Boeing, Rockwell, FMC Corp, Ford-Aerospace, Apple Computer, Teknowledge, IntelliCorp, Syntelligence, Lucid, Inference Corp, Symbolics, LMI, and so on... Many of these firms are marketing hardware and software tools for expert system development, as well as custom system services. And Japan has mounted a long-term, well-funded "Fifth Generation" computing effort to broadly develop knowledge-based systems technology as part of their national economic base of the 1990's.

The AI tide is rising 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 nursery, as well as 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), 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. First among these scientific directions has been a switch in AI's focus from inference procedures to knowledge representation and use. There is now a recognition that the power of problem solvers derives primarily from the knowledge that they contain -- of the elements of the problem domain, of the strategies for solving problems in that domain, and of the forms in which the knowledge is to be acquired. In 1977, Goldstein and Papert of MIT, writing in the journal Cognitive Science, described the change of focus as a "paradigm shift" in AI. This shift was induced largely (though, of course, not exclusively) by the work at SUMEX, beginning with the DENDRAL development in 1965.

Toward the '90s: the Future of SUMEX

Given this setting of success and vitality, 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, with the emergence of powerful single-user workstations and local area networking? The answers remain clear.

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. Our growing knowledge is clearly attained in an incremental fashion; we build today on the results of the past decade, and we will build in the 1990's on the work we undertake today.

At the resource level, there is a growing, diverse, and active AIM research community with intense needs for computing resources to continue its work. Many of these groups still are dependent on the SUMEX-AIM resources. For those who have been able to take advantage of newly developed local computing facilities, SUMEX-AIM provides 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 (previously called the Heuristic Programming Project), make this an ideal setting to integrate, experiment with, and export these methodologies for the rest of the AIM community.

At the beginning, the SUMEX community was small and idea-limited, and the central SUMEX computer facility was an ideal vehicle for the research. Now the community is large, and the momentum of the science is such that its progress is limited by computing power and research manpower. The size and scientific maturity of the SUMEX community has fully consumed the computing resource in every critical dimension -- CPU power, main memory size, address space, and file space -- and has overflowed to decentralized machines of many types. Our projection about the central role of Lisp workstations in AI research and applications has come true dramatically. As we were writing our application five years ago, a few experimental workstations existed in research laboratories and Xerox was laboring over bringing out the first commercial Dolphin. In that short time, Xerox has significantly increased its product line and Symbolics, Lisp Machines Inc., Texas Instruments, and Hewlett Packard have introduced extensive Lisp machine product lines -- at both the low-cost and high-performance extremes of the spectrum. As indicated in the body of this proposal, with NIH and DARPA funding and industrial gifts we have been able to purchase a substantial number of Lisp machines of various types. And much of our work has already been focussed on developing and experimenting with workstation environments for biomedical AI applications. We are fully committed to continuing this line of research for the future hardware thrust of the resource. We will continue our "experimental" approach to these systems, eschewing articles of faith for real experience. We must learn 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.

Our planning axiom for the next period continues to be: the need to accommodate and exploit a *heterogeneity* of computers and peripheral devices. We must maintain a flexible posture with respect to the introduction of new capabilities and changing costs during this continuing revolution. Yet we must choose, while avoiding precipitous decisions. Our plan is conservative, recognizing that there is still a community of national users -- particularly young projects needing seed support prior to obtaining major funding -- who will depend for several years on a central shared resource like the SUMEX mainframe. Since the trend is clear, however, we intend to phase out the role of the central SUMEX machine over the next five years. The existing 2060, with its superb software, will be frozen except for possible minor upgrades (such as in memory) to minimize maintenance costs. It will continue to serve the AIM community during the period of transition, but the costs to SUMEX for its maintenance will decrease linearly until, at the end of the five years, it will no longer be part of the resource. During the phase out period, the 2060 will continue to provide a start-up environment for new projects and will facilitate communication among members of the AIM community. It will also provide us with a "link to the past" -- access to software that is still needed by the community and can be transferred only gradually to the totally distributed computing environment which we anticipate will exist in 1990. The 2060 (plus its satellite 2020 and local VAX's) have been amiable workhorses and, although we do not propose to have SUMEX maintain the smaller mainframe machines into the renewal period, we can not (indeed dare not) do without the 2060 during this period of turbulence. It will have a continuing important role in serving national and local users until an adequate number of workstations gradually become available to all collaborative projects.

On the workstation front, we propose buying a few additional Lisp machines each year to allow our core efforts to stay abreast of the advancing technology. For example, in the first year we plan to buy four of the newly-announced Xerox 6085 machines (these do not even have an 11xx designation yet for the Lisp versions) as the basis for our virtual system development work and the ONCOCIN dissemination research. By the second year, we expect VLSI versions of machines from several companies to choose from and so on through the 5-year term.

These machines will be integrated into the SUMEX local area network and software developed to allow these machines to be more broadly available to local and remote researchers and to cooperate on complex problems. We will enhance the computing environments of these systems to allow users to move off of mainframe systems into increasingly intelligent and supportive surroundings for their work. To facilitate the transfer of software and access to valuable common facilities, the SUMEX complement of equipment will be linked by local digital networks to other major centers of computing at Stanford, the most important of which is the Computer Science Department.

The success of SUMEX is the success of its dedicated and extraordinarily competent faculty and staff. This human resource of SUMEX should not, and will not, be decentralized. In the world of computer systems talent and user-assistance expertise, there are indeed continuing large "economies of scale".

The smoothly operating management structure of SUMEX is one of its joys and victories. We do not plan to fix something that is not broken. We plan that the Executive Committee and the AIM Advisory Committee will continue to function as they now do.

To summarize our goals for the five years that lie ahead:

- Maintain the clear thread through 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 while gradually phasing out the existing DEC-20 machine and focussing new computing resource developments on more effective exploitation of distributed workstations through communication and cooperative computing over transparent digital networking schemes.

- Enhance the computing environments of workstations so that no dependency on central hosts remains and the general mainframe time-sharing systems can be phased out eventually.

- Continue the central staff and management structure, essentially unchanged in size and function, except for the merging of the core part of the ONCOCIN research with the SUMEX resource.

As we add up the budget (flinchingly, we hasten to say), we note that the cost will not be cheap, despite the much-touted fall in the cost of computing. Part of the expense is related to merging the budgets of ONCOCIN and SUMEX, each of which have been separately funded by the Division of Research Resources and are now to be combined in a unified effort. Despite the costs, we believe that we have been conservative; that the scientific community we serve needs these resources; and that by its science and its applications orientation, it has earned them. The scientific work of the SUMEX-AIM community is the quintessence of experimental computer science. It is advancing, and gaining acceptance, beyond expectations. SUMEX serves the nation, not one university or department. We believe that its budget accords well with the national interest and with the scientific interest.

2.1.2. Objectives

2.1.2.1. Resource Goals and Definitions

SUMEX-AIM is a national computer resource with a multiple mission: a) promoting experimental applications of computer science research in artificial intelligence (AI) to biological and medical problems, b) studying methodologies for the dissemination of biomedical AI systems into target user communities, c) supporting the basic AI research that underlies applications, and d) facilitating network-based computer resource sharing, collaboration, and communication among a national scientific community of health research projects. The SUMEX-AIM resource is located physically in the Stanford University Medical School and serves as a nucleus for a community of medical AI projects at universities around the country. SUMEX provides computing facilities tuned to the needs of AI research and communication tools to facilitate remote access, inter- and intra-group contacts, and the demonstration of developing computer programs to biomedical research collaborators.

In the succeeding sections of this proposal, we offer descriptions of these efforts at several levels of detail to meet the needs of reviewers from various perspectives. For this overview, we give only a brief definition of AI and a summary of the aims, background, and present status of our research relative to the requested term of the renewal, the five years beginning August 1, 1986.

What is Artificial Intelligence?

Artificial Intelligence research is that part of Computer Science concerned with symbol manipulation processes that produce intelligent action [1, 56, 61, 69]. Here *intelligent action* means an act or decision that is goal-oriented, is arrived at by an understandable chain of symbolic analysis and reasoning steps, and utilizes knowledge of the world to inform and guide the reasoning.

Placing AI in Computer Science

A simplified view relates AI research with the rest of computer science. The manner of use of computers by people to accomplish tasks can be thought of as a one-dimensional spectrum representing the nature of the instructions that must be given the computer to do its job. At one extreme of the spectrum, representing early computer science, the user supplies his intelligence to instruct the machine precisely *how* to do the job, step-by-step.

At the other extreme of the spectrum, the user describes *what* he wishes the computer to do for him to solve a problem. He wants to communicate what is to be done without having to lay out in detail all necessary subgoals for adequate performance, yet with a reasonable assurance that he is addressing an intelligent agent that is using knowledge of his world to understand his intent, complain or fill in his vagueness, make specific his abstractions, correct his errors, discover appropriate subgoals, and ultimately translate what he wants done into detailed processing steps that define how it should be done by a real computer. The user wants to provide this specification of what to do in a language that is comfortable to him and the problem domain (perhaps English) and via communication modes that are convenient for him (including perhaps speech or pictures).

Progress in computer science may be seen as steps away from that extreme *how* point on the spectrum: the familiar panoply of assembly languages, subroutine libraries, compilers, extensible languages, etc. illustrate this trend. The research activity aimed at creating computer programs that act as *intelligent agents* near the *what* end of the spectrum can be viewed as a long-range goal of AI research.

Expert Systems and Applications

The national SUMEX-AIM resource has in large part made possible a long, interdisciplinary line of artificial intelligence research at Stanford concerned with the development of concepts and techniques for building expert systems [31]. An *expert system* is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. For some fields of work, the knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the expert practitioners of that field.

The knowledge of an expert system consists of facts and heuristics. The *facts* constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in a field. The *heuristics* are the mostly-private, little-discussed rules of good judgment (rules of plausible reasoning, rules of good guessing) that characterize expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses.

Projects in the SUMEX-AIM community are concerned in some way with the application of AI to biomedical research. Brief abstracts of the various projects currently using the SUMEX resource can be found in Appendix D on page 311 and more detailed progress summaries in Section 6 on page 191. The most tangible objective of this approach is the development of computer programs that will be more general and effective consultative tools for the clinician and medical scientist. There have already been promising results in areas such as chemical structure elucidation and synthesis, diagnostic consultation, molecular biology, and modeling of psychological processes.

Needless to say, much is yet to be learned in the process of fashioning a coherent scientific discipline out of the assemblage of personal intuitions, mathematical procedures, and emerging theoretical structure comprising artificial intelligence research. State-of-the-art programs are far more narrowly specialized and inflexible than the corresponding aspects of human intelligence they emulate; however, in special domains they may be of comparable or greater power, e.g., in the solution of structure problems in organic chemistry or in the rigorous consideration of a large diagnostic knowledge base.

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 [41, 11]. And, as projected in our previous application, we are seeing a growing decentralization of computing resources with the emerging technology in microelectronics and a correspondingly greater role for digital communications to facilitate scientific exchange.

Our community building effort is based upon the developing state of distributed computing and communications technology. While far from perfected, these capabilities offer highly desirable latitude for collaborative linkages, both within a given research project and among them. A number of the active projects on SUMEX are based upon the collaboration of computer and medical scientists at geographically separate institutions, separate both from each other and from the computer resource (see for example, the MENTOR and PathFinder projects). Many other projects, once begun using the facilities of the SUMEX-AIM resource, have developed and matured to the point of justifying their own computing resources and now operate independent of, but linked through electronic communications to, the SUMEX-AIM resource. Our network connections and common facilities for user terminals have been indispensable for effective interchanges between community members, workshop coordinations, and software sharing.

2.1.2.2. Specific Aims

The goals of the SUMEX-AIM resource are long term in supporting basic research in artificial intelligence, applying these techniques to a broad range of biomedical problems, developing the methodologies for disseminating AI systems into the biomedical community, experimenting with communication technologies to promote scientific interchange, and developing better tools and facilities to carry on this research.

Toward a More Distributed Resource

In the early 1970's, the initial model for SUMEX-AIM as a centralized resource was based on the high cost of powerful computing facilities and the infeasibility of being able to duplicate them readily. As planned, this central role has already evolved significantly and continues to evolve with the introduction of more compact and inexpensive computing technology now available at many more research sites. At the same time, the number of active groups working on biomedical AI problems has grown and the established ones have increased in size. This has led to a growth in the demand for computing resources far beyond what SUMEX-AIM could reasonably and effectively provide on a national scale. We have actively supported efforts by the more mature AIM projects to develop or adapt additional computing facilities tailored to their particular needs and designed to free the main SUMEX resource for new, developing applications projects. To date, over 10 of the national projects have moved some or all of their work to local sites and several have begun resource communities of their own (see page 116). Thus, as more remotely available resources have become established, the balance of the use of the SUMEX-AIM resource has shifted toward supporting start-up pilot projects and the growing AI research community at Stanford.

Summary of Specific Objectives

Our future goals for the central SUMEX-AIM resource are then guided by:

- The increasingly decentralized character of the resource and community and the need to find ways to maintain the scientific communication and sharing that has characterized SUMEX-AIM work
- The continuing exploration of important new areas of biomedical research in which AI techniques can be effectively applied
- The need for a strong basic research effort to investigate methodologies to attack the many problems still beyond our current AI systems and to develop improved tools to build more complex and effective expert systems
- The growing impact of biomedical AI and the need to find and evaluate ways for effectively disseminating biomedical AI technology into real-world settings.
- The need for computing environments for our research and dissemination work that anticipate the needs of AI applications systems over the next 5-10 years, based on the rapidly changing computing hardware and software technology base

SUMEX-AIM will retain its role as a national experimental laboratory for biomedical AI research with a double thrust -- on the one hand, pursuing the basic research for, experimentation on, and trial dissemination of interesting applications and on the other, anticipating and developing the model computing and community environment in which this work can take place. We will nurture existing and new projects and serve as a communications cross-roads for the now diverse AIM community. We will provide the computing resources and some manpower support for long-term basic AI research activities that promise to illuminate issues relevant to future selected collaborative application areas in biology and medicine. For example, as our detailed plans are presented, you will find threads between our basic research in general patient treatment protocol acquisition, representation, and decision-making tools and our collaborative applications in cancer chemotherapy or hypertension trials. Or between our basic research in blackboard problem-solving frameworks and system architectures and our collaborative application in NMR protein conformation determination. Other basic research areas have even longer term goals for problems we hope to be able to address in the future. Underlying all this work will be the development of the Lisp workstation system and network environment that will facilitate these research results and that we feel will become the routine computing environment of the next decade.

In all of this, SUMEX will be both a working laboratory for selected projects within our computing and manpower capacity limits and a source and repository for software and ideas for a broader remote community. We will become an increasingly distributed community resource with heterogeneous computing facilities tethered to each other through various communications media. Many of these machines will be located physically near the projects or biomedical scientists using them. We retain our sincere commitment to our national community of projects. But, inevitably their needs will be met more and more by local facilities and our plans as a resource for the next term place greater emphasis than in the past on supporting the growing Stanford community of AIM collaborations and projects and on developing and integrating model systems at Stanford that can be emulated elsewhere for AIM community needs.

Even with more distributed computing resources, the central resource will continue to play an important role for the next term as a communication crossroads and as a focus for our active dissemination efforts. A key challenge will be to maintain the scientific community ties that grew naturally out of the previous co-location within a central facility.

The following outlines the specific objectives of the SUMEX-AIM resource during the follow-on five year period. Note that these objectives cover only the resource nucleus; objectives for individual collaborating projects are discussed in their respective reports in Section 6. Specific aims are broken into five categories: 1) Core Research and Development, 2) Collaborative Research, 3) Service and Resource Operations, 4) Training and Education, and 5) Dissemination.

1) Core Research and Development

SUMEX funding and computational support for core research is complementary to similar funding from other agencies (see page 105) 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, MRS, SOAR, 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 computing resources under the control of the AIM management committees. There will be no "fee for service" charges for community members.

- 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 TYMNET 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.

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 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.

2.1.2.3. Resource Scope

The SUMEX-AIM resource has been from its inception a national experimental resource for biomedical AI with a scope that is carefully defined. Within its limited manpower and computational resources, its focus has been on experiments in new and varied biomedical applications of AI, assisting new research groups in biomedical AI get started, exploring ways to disseminate AI systems into biomedical user communities, supporting relevant basic AI research, and facilitating scientific communications and community sharing. The SUMEX-AIM user community comprises projects from many biological and medical disciplines, ranging from chemistry to molecular biology to clinical medicine to cognitive psychology, and represents collaborations between computer and biomedical scientists from many parts of Stanford University and other universities around the country. The development of this diverse community of projects has both justified the cost of and made effective use of SUMEX-AIM computational and communication facilities at Stanford and elsewhere in our resource community. In its resource role, SUMEX has intentionally limited its production computational capacity to meet the needs of national start-up projects and Stanford research groups, while encouraging self-sufficient community members to develop resources to meet their own computing needs. This has allowed us to provide a level of support for on-going projects and to concentrate most of our efforts on experiments with integrating emerging hardware and software technologies that will be the vehicles of future biomedical AI systems. The results of these experiments are widely disseminated and help other groups through example and direct export of software and ideas.

2.1.2.4. Significance to Biomedicine

Artificial intelligence is the computer science of representations of symbolic knowledge and its use in symbolic inference and problem-solving processes. There is a certain inevitability to this branch of computer science and its applications, in particular, to medicine and biosciences. The cost of computers will continue to fall drastically during the coming two decades. As it does, many more of the practitioners of the world's professions will be persuaded to turn to economical automatic information processing for assistance in managing the increasing complexity of their daily tasks. They will find, from most of computer science, help only for those problems that have a mathematical or statistical core, or are of a routine data-processing nature. But such problems will be relatively rare, except in engineering and physical science. In medicine, biology, management, indeed in most of the world's work, the daily tasks are those requiring symbolic reasoning with detailed professional knowledge. The computers that will act as *intelligent assistants* for these professionals must be endowed with symbolic reasoning capabilities and knowledge.

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.

The knowledge that must be used is a combination of factual knowledge and heuristic knowledge. The latter is especially hard to obtain and represent since the experts providing it are mostly unaware of the heuristic knowledge they are using. Medical and scientific communities currently face many widely-recognized problems relating to the rapid accumulation of knowledge, for example:

- codifying theoretical and heuristic knowledge
- effectively using the wealth of information implicitly available from textbooks, journal articles and other practitioners
- disseminating that knowledge beyond the intellectual centers where it is collected
- customizing the presentation of that knowledge to individual practitioners as well as customizing the application of the information to individual cases

We believe that computers are an inevitable technology for helping to overcome these problems. While recognizing the value of mathematical modeling, statistical classification, decision theory and other techniques, we believe that effective use of such methods depends on using them in conjunction with less formal knowledge, including contextual and strategic knowledge.

Artificial intelligence offers advantages for representing and using information that will allow physicians and scientists to use computers as intelligent assistants. In this way we envision a significant extension to the decision-making powers of specific practitioners without reducing the importance of those individuals in that process.

Knowledge is power, in the profession and in the intelligent agent. As we proceed to model expertise in medicine and its related sciences, we find that the power of our programs derives mainly from the knowledge that we are able to obtain from our collaborating practitioners, not from the sophistication of the inference processes we observe them using. Crucially, the knowledge that gives power is not merely the knowledge of the textbook, the lecture and the journal, but the knowledge of *good practice*--the experiential knowledge of *good judgment* and *good guessing*, the knowledge of the practitioner's art that is often used in lieu of facts and rigor. This heuristic knowledge is mostly private, even in the very public practice of science. It is almost never taught explicitly, is almost never discussed and critiqued among peers, and most often is not even in the moment-by-moment awareness of the practitioner.

Perhaps the the most expansive view of the significance of the work of the SUMEX-AIM community is that a methodology is emerging for the systematic explication, testing, dissemination, and teaching of the heuristic knowledge of medical practice and scientific performance. Perhaps it is less important that computer programs can be organized to use this knowledge than that the knowledge itself can be organized for the use of the human practitioners of today and tomorrow.

Evidence of the impact of SUMEX-AIM in promoting ideas such as these, and developing the pertinent specific techniques, has been the explosion of interest in medical artificial intelligence and the specific research efforts of the SUMEX community. In SUMEX's second decade, we have found that the small community of researchers that characterized the AIM field in the early 1970's has now grown to a large, accomplished, and respected research community. The American Association for Artificial Intelligence (AAAI), the principal scientific membership organization for the AI field, has 7000 members, over 1000 of whom are members of the medical special interest group known as the AAAI-M. This subgroup was founded by members of the SUMEX-AIM community who were active in AAAI and is the only active subgroup in the Association. The organization distributes semiannual newsletters on medical AI and provides a focus for cosponsoring relevant medical computing meetings with other societies (such as the American Association for Medical Systems and Informatics -- AAMSI). Medical AI papers are prominently featured at both medical computing and artificial intelligence meetings, and artificial intelligence is now routinely featured as a specific subtopic for specialized sessions at medical computing and other medical professional meetings. For example, members of the AIM community have represented the field to physicians at the American College of Pathology and American College of Physicians meetings for the last several years. A mere decade ago, the words "artificial intelligence" were never uttered at such conferences. The growing interest and recognition are largely due to the activities of the SUMEX-AIM community.

Another indication of the growing impact of the SUMEX-AIM community is its effect on medical education. For reasons such as those outlined above, there is an increasing recognition of the need for a revolution in the way medicine is taught and medical students organize and access information. Computing technology is routinely cited as part of this revolution, and artificial intelligence (and SUMEX-AIM research) generally figures prominently in such discussions. 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; reports of all four organizations have specifically cited the role of artificial intelligence techniques in future medical practice and have used SUMEX-AIM programs as examples of where the technology is gradually heading.

In summary, the logic which mandates that artificial intelligence play a key role in enhancing knowledge management and access for biomedicine -- a logic in which we have long believed -- has gradually become evident to much of the biomedical community. We are encouraged by this increased recognition, but humbled by the realization of the significant research challenges that remain. Our goals are accordingly both scientific and educational. We continue to pursue the research objectives that have always guided SUMEX-AIM, but must also undertake educational efforts designed to inform the biomedical community of our results while cautioning it about the challenges remaining.

2.1.3. Background

Beginning in the mid-1960's with DENDRAL, a project focused on applications of artificial intelligence to experiments in modeling scientific inference in biomolecular structure characterization problems [43], the Stanford Knowledge Systems Laboratory (formerly named the Heuristic Programming Project, see Appendix A) has pioneered in expert systems research with funding support from NIH, ARPA, NSF, and NASA and other government and private sources. Much of the early DENDRAL computation work was done on the ACME IBM 360/50 interactive computing resource at Stanford, which was funded by the NIH Biotechnology Resources Program between 1965 and 1973. This system, while an excellent experiment in interactive medical computing, could not provide the symbolic computing resources needed for AI research. Such resources were not available from other sources either since the system hardware and software requirements for AI research (for example, address space, memory size, languages and debugging support, and interactive facilities) surpass the services customarily offered in academic or commercial computing facilities. With the success of DENDRAL by the early 1970's and the start of experiments in other application areas such as clinical medicine, chemical synthesis, learning, and cognitive psychology, a general need for state-of-the-art AI computing resources became manifest. Because no single project could justify funding for its own computing facility of the needed magnitude, we were led to formulate a shared community solution that was to have far-reaching impact, both in the support of biomedical AI research and as an experiment in electronic collaboration among scientists. Since 1973, SUMEX-AIM has developed as a national resource for applying AI techniques to a broad range of biomedical research problems.

Funding of the SUMEX-AIM resource from the NIH Biomedical Research Technology Program (formerly Biotechnology Resources Program) began in December 1973 for a five year period. Prof. Joshua Lederberg was Principal Investigator and Prof. Edward A. Feigenbaum was co-Principal Investigator. The major hardware, a DEC KI-10 system running the experimental TENEX operating system, was delivered and accepted in April 1974, and the system became operational for users during the summer of 1974. In 1977, we applied for a five-year renewal grant to continue our national research effort. We received a recommendation for approval of the five year period from the study section but this was reduced to three years following Professor Lederberg's decision in early 1978 to accept the presidency of The Rockefeller University. The principal investigator role passed easily to Prof. Feigenbaum, then Chairman of the Stanford Computer Science Department, based upon his long-time involvement with the project and close collaboration with Prof. Lederberg. The highly interdisciplinary spirit of SUMEX was retained with very close ties to the Stanford Medical School through Drs. E. H. Shortliffe (then co-Principal Investigator of SUMEX) and S. N. Cohen. At the end of that 3-year term, we applied for and were awarded a third renewal for the SUMEX-AIM resource for 5 years starting in August 1981, under Professors Feigenbaum and Shortliffe. This winter, with his appointment to a tenured faculty position at Stanford and with his physical and administrative proximity to the SUMEX resource in the Department of Medicine, Dr. Shortliffe took over as Principal Investigator of SUMEX and Professor Feigenbaum again became co-Principal Investigator

Although the 12 years of support the SUMEX-AIM resource has received is long by some standards, it is short in terms of the time needed to develop the discipline of artificial intelligence and to realize the potential of its applications in biomedicine. The existence of SUMEX-AIM and its support by NIH has been crucial to the substantial progress made to date. It is hardly long enough for a conclusive determination of the long term impact of this work but we can fairly take pride in the scientific success of SUMEX-AIM as a community and in the success of the SUMEX-

AIM model as a resource. Beginning with 5 projects in 1973, over 35 research projects have started within the SUMEX-AIM community and, after initial nurturing, 9 have developed independent computing resources of their own and now operate as autonomous projects. More than a dozen books describing the results of community work have been published since 1980. And as indicated earlier, increased training and the use of computing in general, including programs centered on symbolic computation, are being advocated for medical education and research. Finally, significant progress has been made in starting the commercialization of AI technology, based to a significant extent on the success of early research in the SUMEX-AIM community.

On the resource management side, we take pride in the diligence and technical competence with which we have responded to the community responsibilities mandated by the terms of our grants [51]. Good will and common purpose are of course the indispensable ingredients for an effective community resource, and we are grateful to have been able to offer this service in a congenial framework, and at the same time to be able to support our local computing research needs. The character of the SUMEX resource has changed with the evolving computer and communications technology on which it is based. Starting with fully centralized hardware and distributed research groups in 1974, the community (research groups and computing resources) is now highly distributed. This change is essential to the technical vitality of the on-going work and to ensuring the availability of computing resources that will be the means for disseminating AI programs to biomedical researchers and practitioners.

The present renewal application is therefore written from a perspective of having built a significant community of active biomedical AI research projects and of entering a new phase of our research to integrate and exploit exciting computer technologies that will have a profound effect on the development and export of practical medical AI programs. As discussed in the sections describing the individual projects (see Section 6), many of the computer programs under development by these groups are maturing into tools increasingly useful to the respective research communities. The demands from innovative new core research work and for production-level use of these programs has long ago surpassed the capacity of the present SUMEX facility and has raised important issues of how such software systems can be developed in effective research environments and then optimized for production environments, exported, and maintained.

2.1.4. Resource Progress

This progress summary covers only the resource nucleus. Objectives and progress for individual collaborating projects are discussed in their respective reports in Section 6. In particular, progress in the current ONCOCIN resource-related research project for Studies in the Dissemination of Consultation Systems, which will be merged with SUMEX in the renewal period, is reported there. Longer term goals for the ONCOCIN core research work over the period of this renewal application are discussed under the *Planned Resource Activities* section of this proposal. 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.

2.1.4.1. Summary of Prior Goals

The following summarizes SUMEX-AIM resource objectives as stated in the proposal for the on-going five-year grant, begun on August 1, 1981, and provides the backdrop against which specific progress is reported. These project goals are presented in the three categories used in the previous proposal: 1) resource operations, 2) training and education, and 3) core research.

1) Resource Operations

- Maintain the vitality of the AIM community by continuing to encourage and explore new applications of AI to biomedical research and improving mechanisms for inter- and intra-group collaborations and communications. User projects will fund their own manpower and local needs; will actively contribute their special expertise to the SUMEX-AIM community; and will receive an allocation of computing resources under the control of the AIM management committees. There will be no "fee for service" charges for community members.

- Provide effective computational support for AIM community goals, including efforts to improve the support for artificial intelligence research and new applications work; to develop new computational tools to support more mature projects; and to facilitate testing and research dissemination of nearly operational programs. We will continue to operate and develop the existing KI-10/2020 facility as the nucleus of the resource. We will acquire additional equipment to meet developing community needs for more capacity, larger program address spaces, and improved interactive facilities. New computing hardware technologies becoming available now and in the next few years will play a key role in these developments and we expect to take the lead in this community for adapting these new tools to biomedical AI needs. We planned the phased purchase of two VAX computers to provide increased computing capacity and to support large address space LISP development, a 2 GByte file server to meet file storage needs, and a number of single-user "professional workstations" to experiment with improved human interfaces and AI program dissemination.

- 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 TYMNET connections for at least the near term and will actively explore other advantageous connections to new communications networks and to dedicated links.

2) **Training and Education**

- Provide community-wide support and work to make resource goals and AI programs known and available to appropriate medical scientists. Collaborating projects are responsible for the development and dissemination of their own AI programs.

- Provide documentation and assistance to interface users to resource facilities and programs and continue to exploit particular areas of expertise within the community for developing pilot efforts in new application areas.

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

- Support workshop activities, including collaboration with the Rutgers Computers in Biomedicine resource on the AIM community workshop and with individual projects for more specialized workshops covering specific application areas or program dissemination.

3) **Core Research**

- Explore basic artificial intelligence research issues and techniques, including knowledge acquisition, representation, and utilization; reasoning in the presence of uncertainty; strategy planning; and explanations of reasoning pathways, with particular emphasis on biomedical applications.

- Support community efforts to organize and generalize AI tools that have been developed in the context of individual application projects. This will include work to organize the present state-of-the-art in AI techniques through the AI Handbook effort and the development of practical software packages (e.g., AGE, EMYCIN, UNITS, and EXPERT) for the acquisition, representation, and utilization of knowledge in AI programs.

2.1.4.2. Progress Highlights

In this section we summarize highlights of SUMEX-AIM resource activities over the past 4 years, focusing on the resource nucleus.

- 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 freeing up SUMEX resources for newer projects. Nine projects now operate on their own facilities, including three that have become BRTP resources in their own right. Nine projects in the community have completed their research goals and their staffs have moved on to new areas.

- SUMEX user projects have made good progress in developing and disseminating effective consultative computer programs for biomedical research. These performance programs provide expertise in analytical biochemical analyses and syntheses, clinical diagnosis and decision-making, molecular biology, and various kinds of cognitive and affective psychological modeling. We have worked hard to meet their needs and are grateful for their expressed appreciation (see Section 6).

- We have made significant strategic improvements to the SUMEX-AIM computing environment in order to optimize computing support for the community. These developed in ways somewhat different from the initially projected plan. The DEC VAX computer did not prove to be an effective machine for running Lisp [45], while Lisp workstations have in fact become available from a number of vendors as tentatively expected at the time of our proposal (first Xerox, then Symbolics and LMI, and more recently Hewlett-Packard and Texas Instruments). Thus, rather than augmenting our mainframe resources with the purchase of large address space VAX's, we upgraded the KI-TENEX system to a DEC 2060 and at the same time, began moving aggressively toward a Lisp workstation-based research environment, with the approval of an ad hoc site visit group. We did secure VAX capabilities for our community by means of access to an 11/780 purchased under DARPA funding. We made an initial purchase of Xerox Dolphins with NIH funding and subsequently added more Xerox and Symbolics machines with NIH and DARPA funding and with industrial gifts. Because of the broad mix of research in the SUMEX-AIM community, no single workstation vendor can meet our needs so we have undertaken long-term support of a heterogeneous computing environment, incorporating many types of machines linked through multiprotocol Ethernet facilities.

- We have continued the dissemination of SUMEX-AIM technology through various media. We have distributed various AI software tools to many research laboratories, including over 200 combined copies of the GENET, EMYCIN, AGE, MRS, SACON, GLISP, and BB-1 systems. Several of our software systems have been adapted as commercial AI tools such as the Teknowledge S.1 and M.1 systems derived from EMYCIN, the Texas Instruments Personal Consultant system derived from EMYCIN, and the IntelliCorp KEE system derived from UNITS. We have also prepared video tapes of some of our research projects including ONCOCIN and an overview tape of Knowledge Systems Laboratory work.

- 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 (see page 109). These books have included the three-volume set of the *Handbook of Artificial Intelligence*, edited by Barr, Cohen, and Feigenbaum; a book on *Readings in Medical Artificial Intelligence: The First Decade* by Clancey and Shortliffe; and a book on *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* by Buchanan and Shortliffe.

- We completed the GENET project, begun in 1980 as a collaboration between the MOLGEN investigators and SUMEX, to make a set of DNA sequence analysis computing tools available to a national community of molecular biologists. This was an experiment in using a SUMEX-like resource to disseminate sophisticated software tools to a computer-naive community and proved extremely successful. GENET served over 300 molecular biologists before being phased out in early 1983. Subsequently, a new resource called BIONET has been funded by NIH at IntelliCorp to provide routine service of the type pioneered by SUMEX/GENET.

- A program in Medical Information Sciences was begun at Stanford in 1983 under Professor Shortliffe as Director. A group of faculty from the Medical School and the Computer Science Department argued that research in medical computing has historically been constrained by a lack of talented individuals who have a solid footing in both the medical and computer science fields. The specialized curriculum offered by the new program is intended to overcome the limitations of previous training options. It focusses 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 feasibility of this program resulted in large part from the prior work and research computing environment provided by the SUMEX-AIM resource. Over 20 PhD and MS trainees will be enrolled in the fall of 1985. It has been awarded post-doctoral training support from the National Library of Medicine, received an equipment gift from Hewlett-Packard, and has received additional industrial and foundation grants for student support.

- We made significant progress in core AI research. In the area of knowledge representation, work was done on the representation of explicit strategy knowledge, temporal knowledge, causal knowledge, and knowledge in logic-based systems. In the area of architectures and control, we worked on a new implementation of a blackboard architecture with explicit control knowledge. Under knowledge acquisition studies, three PhD theses were completed covering experiments in learning by induction, by analogy, and learning from partial theories. In the area of knowledge utilization, results include work on reasoning with uncertainty and using counterfactual conditionals. We continued work on a number of existing tools for expert systems and on building new ones such as the BB1 system. And finally, significant work was done on the inference of user models, skeletal planning, defining a taxonomy of diagnostic methods, and reasoning with causal models.

- We have continued the core development of the SUMEX facility hardware, software, and 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 machines based on

MC-68000 microprocessors were developed to link our environment together and are now the standard system used in the campus-wide Stanford University network. We developed a gateway interface between Apple equipment (e.g., the Macintosh and Lisa) and EtherNet hosts that is now in wide use at universities around the country. We have developed many other software packages to enhance the computing environments of the Lisp workstations and to link them to other hosts and servers on our networks.

The following sections then give more detail about SUMEX-AIM core resource activities since the last grant award.

2.1.4.3. Resource Equipment Details

The SUMEX-AIM core facility, started in March 1974, was built around a Digital Equipment Corporation (DEC) KI-10 computer and the TENEX operating system which was extended locally to support a dual processor configuration. Because of the operational load on the KI-10's, in the late 1970's, we had added a small DEC 2020 system (see Figure 2) to support more dedicated testing of systems like ONCOCIN and Caduceus and for community demos. This facility provided a superb base for the AI mission of SUMEX-AIM through 1982. Its interactive computing environment, its AI program development tools, and its network and interpersonal communication media were unsurpassed in other machine environments. Biomedical scientists found SUMEX easy to use in exploring applications of developing artificial intelligence programs for their own work and in stimulating more effective scientific exchanges with colleagues across the country. Coupled through wide-reaching network facilities, these tools also give us access to a large computer science research community, including active artificial intelligence and system development research groups.

The Heterogeneous Computing Environment

In the renewal for the current grant period, both an augmentation of the central resource in terms of address space and capacity and exploratory work with Lisp workstations were planned. The Initial Review Group recognized in their special study section report the importance of optimizing the timing of our planned hardware acquisitions to coordinate community needs with the availability of important technological developments in vendor-supported systems. They recommended in their report that we be allowed considerable flexibility as to phasing of equipment purchases within the 5-year renewal period.

We had initially planned to purchase a large VAX in 1981 and later, our first Lisp workstations. However, we speeded our push toward workstations for several reasons. The state of VAX Lisp implementations and projections of their performance were very discouraging (a study of the VAX InterLisp implementation was done at the time as documented in [45]). And the first Xerox InterLisp Dolphin workstations were available for delivery after the summer of 1981. These machines were the prototypes on which research toward adapting expert AI systems for the interactive workstation environment could begin. So, we purchased 5 Dolphins for the fall of 1981 and, in order not to delay non-Lisp SUMEX-AIM work involving VAX machines, we were able to arrange shared access to a VAX 11/780 funded by ARPA to support Heuristic Programming Project research. One of the Dolphins we purchased was loaned for several years to the Rutgers Computers in Biomedicine resource for experimental work.

We continued to evaluate strategies and alternatives for planned system configuration development. In particular, we had a chance to gain experience with the Dolphin InterLisp machines and the shared VAX, reassess the role of the dual KI-TENEX system, and reach a consensus about what the long term configuration of the SUMEX-AIM facility should be. This was validated by an ad hoc study section review in 1982. In summary, it was decided that the best resource configuration for the coming decade would be a shared central machine coupled through a high-performance network to growing clusters of personal workstations. The central machine should be an extended addressing TOPS-20 machine and the workstations will be chosen from the viable products available and scheduled for announcement.

The concept of the individual workstation, especially with the high-bandwidth graphics interface, proved ideal. Both program development tools and facilities for expert system user interactions were substantially improved over what is possible with a central time-shared system. The main shortcomings of these systems were their processing speed and cost, but the prospect of other workstations to be available from Xerox, Symbolics, LMI, HP, and others reassured us that these were the right choices for AI system in the long term. Still, at the time, it was not possible to equip very much of the SUMEX-AIM community with individual workstations.

Upgrade of the KI-10's to a 2060

Meanwhile, on the mainframe front, given the continued need for a central machine, the poor Lisp performance of the VAX, and the increasingly untenable difficulties in maintaining the KI-TENEX system, we decided it is time to retire the KI-10's and upgrade them to the then (1982) more modern DEC 2060 TOPS-20 system. This would free our systems staff to concentrate on more productive development efforts for the community such as work related to professional workstations and compatible Lisp support. The 2060 had a processing capacity of 2-3 times that of the dual KI-TENEX system, badly needed for our community, and it was more compact, reliable, and maintainable. Pending the arrival of more cost-effective and generally-available Lisp workstations, this would allow us to continue support for the SUMEX-AIM community at large and to provide facilities for new AI efforts.

In late 1982, we implemented the upgrade. The purchase price of the DECsystem 2060 reflected a substantial price reduction based on an external research grant from Digital Equipment Corporation to the Heuristic Programming Project in exchange for access by DEC to the AI software systems and knowledge-based systems expertise developed by the HPP. The remainder of the system was funded jointly by NIH and DARPA. The system configuration is shown in Figure 1. Of course, the transfer of service required a substantial investment of hardware engineering effort as all of the local line and network connections had to be changed over. This was all effected invisibly to the user community by running the old KI-TENEX and the new 2060 systems in parallel for more than a month.

Using DARPA funding, we also made some upgrades to the shared VAX 11/780 which was initially purchased by ARPA for HPP research as well as work in network graphics and VLSI design. The configuration of this machine is shown in Figure 3. In 1983, we augmented the machine by adding 2 Mbytes of memory and expanding the file system with a DEC RP07 disk drive (512 Mbytes). Approximately 60% of the machine is allocated for HPP and SUMEX use.

The overall facility model then became the central shared 2060, 2020, and VAX 11/780 systems surrounded with growing numbers of workstations and intercoupled by a local area network.

Additional Workstations

After the purchase of the 5 experimental Dolphin workstations, much work went into their development by Xerox, based on feedback and interactions with groups such as ours using them for AI applications. Performance of the Dolphins improved substantially based largely on improved microcoding of frequently used primitives and facilities. The initial optimizations of the Dolphin microcode were based on work at Xerox observing their own programs running. When the Dolphin was exposed to other AI systems such as ours, it became clear that additional improvements were necessary and were implemented, including enhanced performance for CONS operations, function calls, disk management, garbage collection, and other areas. Improvements in individual areas of performance ranged from factors of 2 to 10.

By 1983, other contenders were entering the Lisp workstation market in addition to Xerox. Because work in the HPP and the SUMEX-AIM community draws heavily on both Interlisp and the derivatives of MIT's MacLisp, we broadened our workstation experiments into both areas.

With NIH funding in 1983, we purchased 6 Xerox 1108 workstations (Dandelions) and in 1984, 3 Xerox 1109's (DandeTigers). With DARPA funding we purchased 2 Xerox 1108's and 1 1132 (high-performance Dorado) in 1984. In early 1985, the ONCOCIN group received a grant from Xerox of 13 1108's and additional printing and file server equipment. These machines represent the second generation of Xerox Lisp workstations and include significantly higher performance and functionality.

With DARPA funding in 1983 we bought a Symbolics LM-2 running the ZetaLisp system. In 1984, we added 3 Symbolics 3600's and a 3670 and in early 1985, another 3670 -- all with DARPA funding. We are also planning the purchase of additional workstations in the near term with DARPA funding.

Local Area Network Server Hardware

Since the late 1970's, we have been developing a local, high-speed Ethernet environment to provide a flexible basis for planned facility developments and the interconnection of a heterogeneous hardware environment. Our development of Ethernet facilities has been guided by the goals of providing the most effective range of services for SUMEX community needs while remaining compatible with and able to contribute to and draw upon network developments by other groups, dating back to the early 3 Mbit/sec Ethernet given to Stanford and several other universities by Xerox. We now support both 3 and 10 Mbit/sec Ethernets (see Figure 5) running numerous protocols and extended geographically throughout the SUMEX-AIM and related Stanford research groups. This network is the "glue" that holds the rest of the computing environment together and consists of numerous servers such as gateways and servers for terminal access, file storage and retrieval, and laser printing.

In the early phases, a substantial amount of special hardware was developed by our group for network interfaces including a high-performance direct memory access interface for the dual KI-TENEX system and a serial phase decoded UNIBUS interface that are used on our DEC 2020, VAX's, and early PDP-11 gateways and TIP's. The KI Ethernet interface served well for a period until we upgraded the system to a 2060, at which time we installed the 2060 mass bus EtherNet interface designed and built by the Stanford Computer Science Department. Our KI-10 interface is still seeing service in connecting another KI-10 system (Institute for Mathematical Studies in the Social Sciences) to the net.

Hardware for Gateways and TIP's

As we evolved a more complex network topology and decided to compartmentalize the overall Stanford internet to avoid electrical interactions during development and to facilitate different administrative conventions for the use of the various networks, we developed gateways to couple subnetworks together. These first used PDP-11/05 hardware and then Motorola MC-68000 systems as they became available.

Similarly, we designed gateway between Apple equipment such as the new Macintosh terminal, that may play a role in our future virtual graphics work (see page 162), and EtherNet using a MC-68000 gateway and a locally-designed Apple Bus to Multibus EtherNet interface. This system incorporates an 8530 Zilog chip to communicate with the Apple Net and software to manage the protocol packaging.

We also developed a MC-68000 terminal interface processor (TIP) to provide terminal access to network hosts and facilities. It is basically a machine that has a number of terminal lines and a network interface and software to manage the establishment of connections for each line and the flow of characters between the terminal and host. It can handle up to 32 lines. Both of these systems are now widely used throughout the Stanford network.

File Server Hardware

The development of an EtherNet file server was an integral part of our council-approved equipment plan with further expansions approved for later years. With joint NIH and DARPA funding, we were able to take advantage of an exceptional offer by Digital Equipment Corporation, through their corporate external research sponsorship program to DARPA contractors (the HPP), to purchase two VAX 11/750 machines as the processor part of our file servers. In the initial file server configurations, we also bought Fujitsu Eagle 450 MByte disks and controllers (one each from Systems Industries and Emulex) with one 800/1600 BPI tape unit for long term archives, and one 300 Mbyte removable pack drive for cyclic backups.

Other Network Hardware

We have developed numerous local network connection systems that have taken advantage of existing cabling rather than invest in expensive trenching and recabling. For example, in The Heuristic Programing Project (HPP) move to 701 Welch road, a high-performance network link to other SUMEX and campus network facilities was essential. Several communication schemes for establishing a reliable and relatively fast link were considered, including microwave, infrared laser, direct ethernet (by trenching and placing a direct ethernet cable), telephone company T1 service and others. All of these would have involved high cost and so we developed a communication link using bare copper telephone pair already in place. The wire distance between the HPP Welch Road location and the SUMEX machine room in the Medical Center is approximately 2000 ft. Utilizing high capacity differential drivers and ultra high speed, high sensitivity receivers, a half-duplex transceiver was developed for plain copper twisted pair that achieved error-free transmission at 1.25 Mbits/sec in each direction, utilizing Manchester data encoding. This communication link has been in operation for well over a year now without any appreciable down time or noticeable error rate or data delays.

In addition to the normal continuous flow of maintenance problems, we have reconnected the very reliable line printer from the old KI-TENEX system to the 2060. This required substantial modification of the printer controller to adapt to the different 2060 bus signal standards. We have also installed lots of communications equipment, including dial-in and -out modems and laser printer connections.

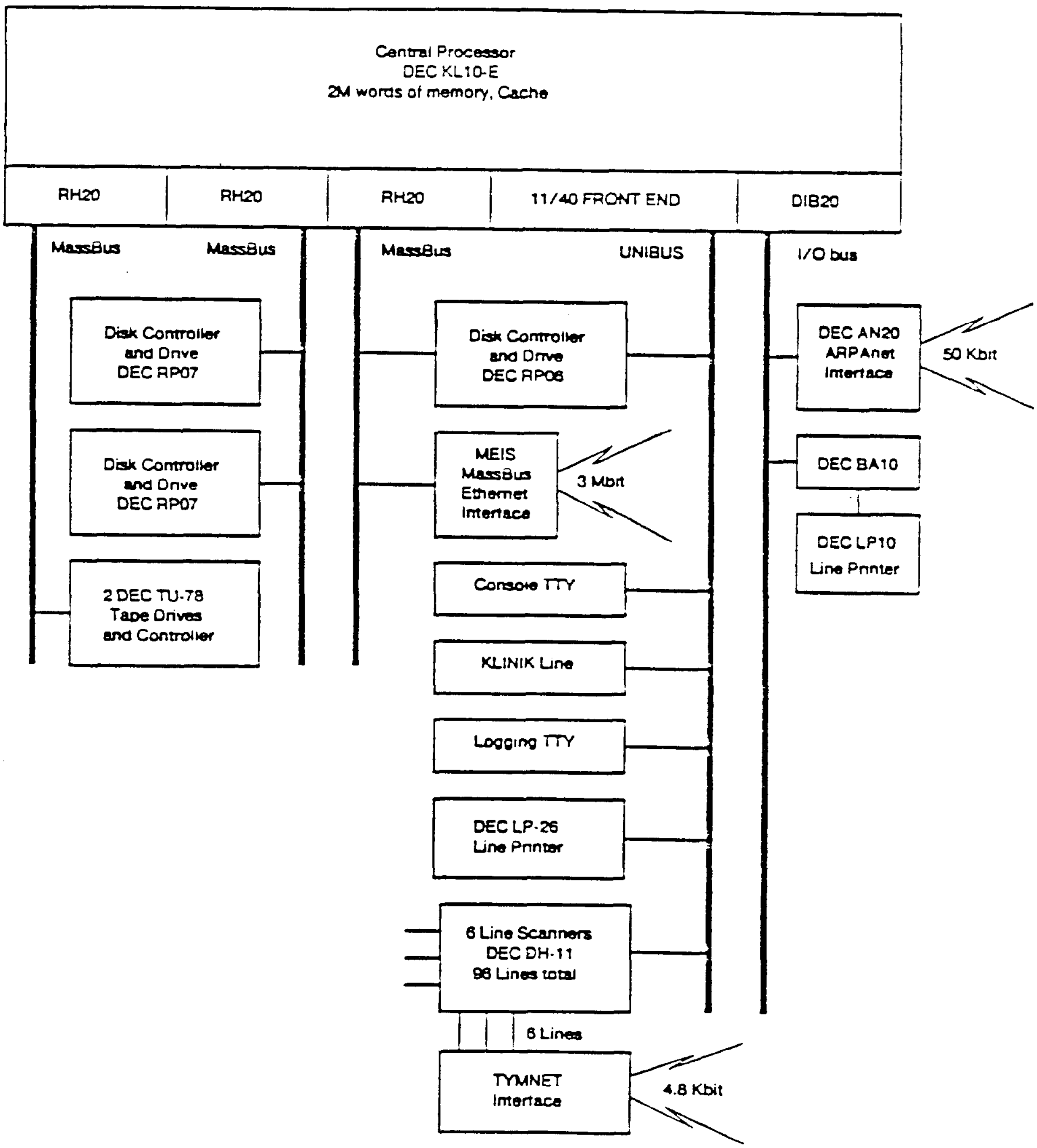

Figure 1: SUMEX-AIM DEC 2060 Configuration

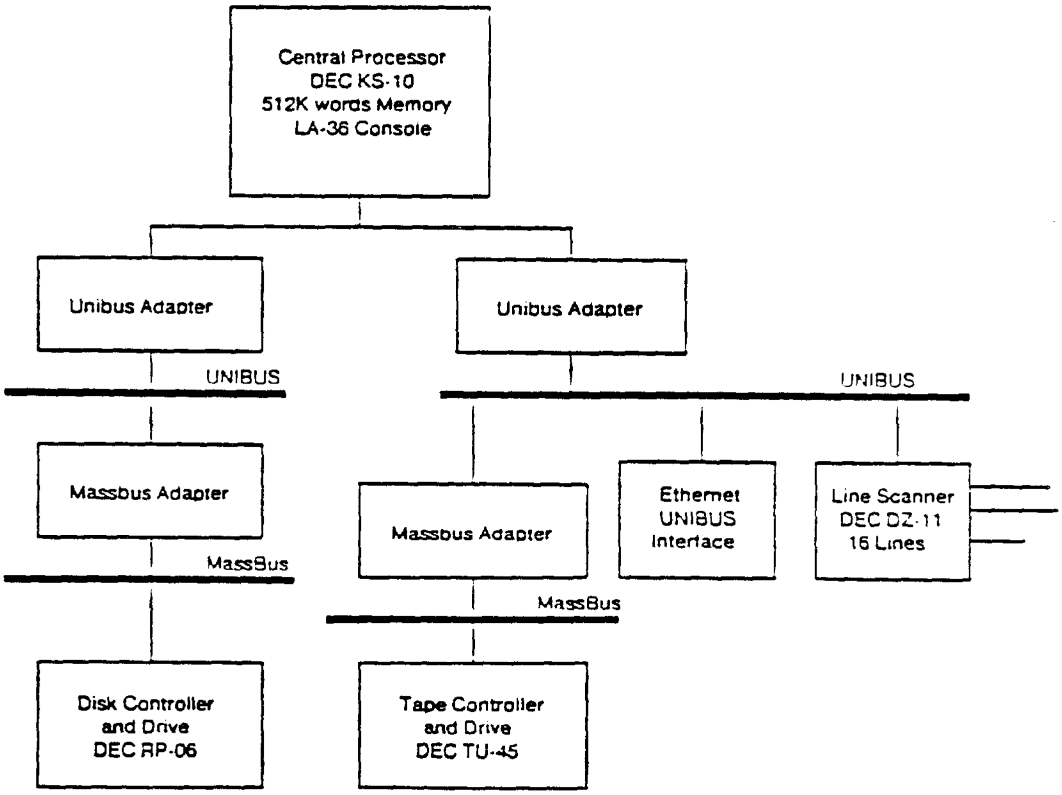

Figure 2: SUMEX-AIM DEC 2020 Configuration

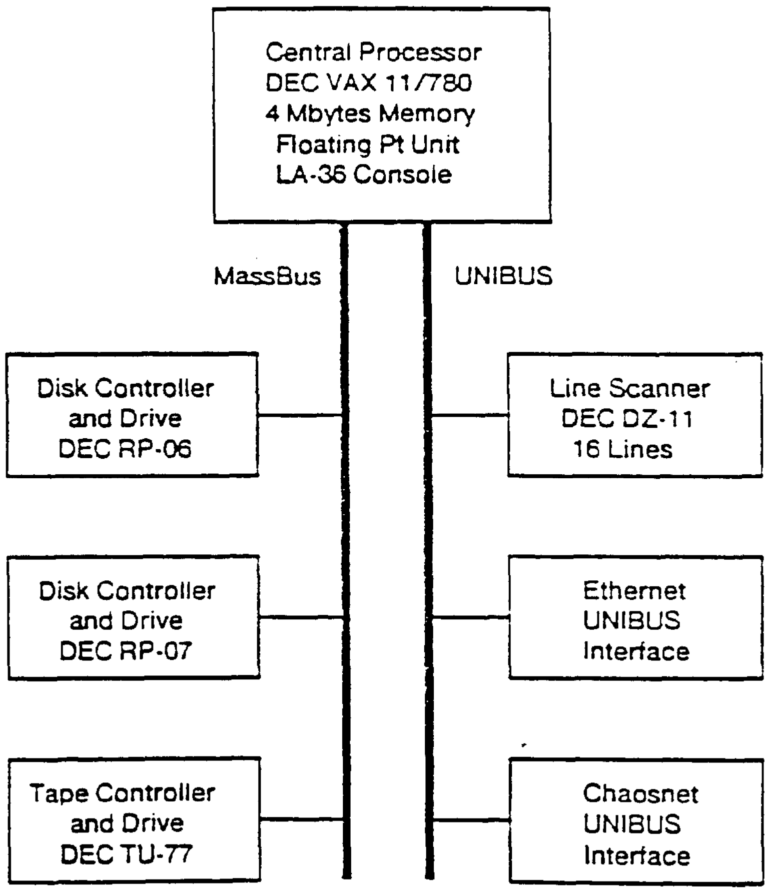

Figure 3: SUMEX-AIM Shared DEC VAX 11/780 Configuration

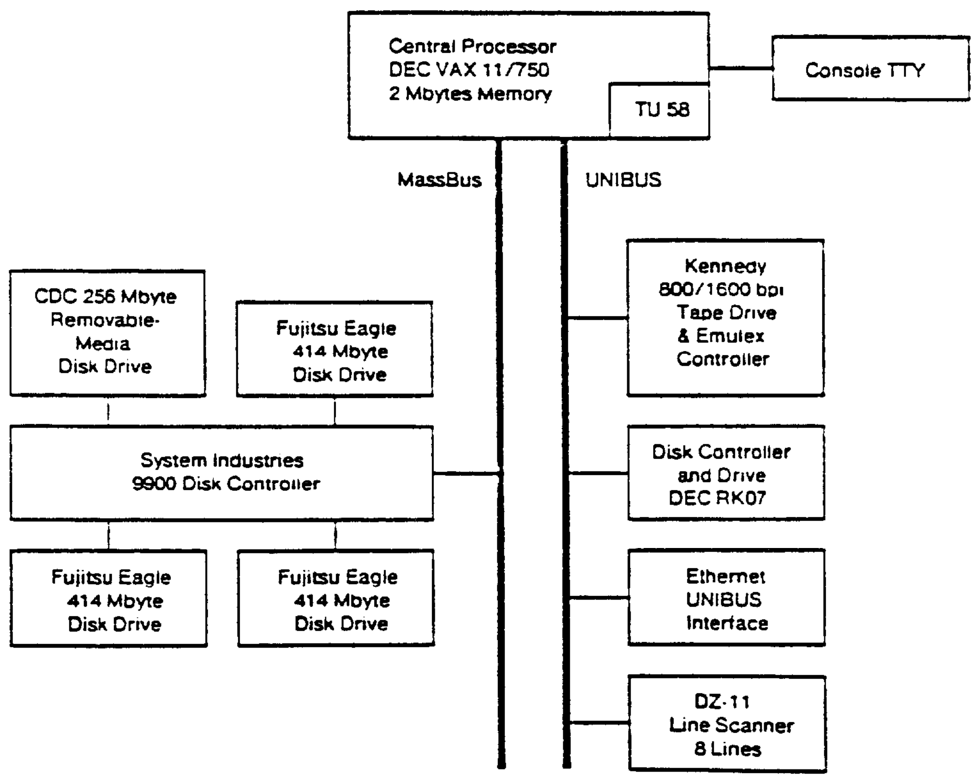

Figure 4: SUMEX-AIM File Server Configuration

2.1.4.4. Core System Development

Operating System Software

The various hardware elements of the SUMEX-AIM computing environment require the development and support of the operating systems that provide the interface between user software and the raw computing capacity. In addition to performance and relevance to AI research, much of our strategy for hardware selection has been based on being able to share development of the operating systems for our research among a large computer science community. This includes the mainframe systems (TOPS-20 and UNIX) and the workstation systems. Following are some highlights of recent system software developments.

TOPS-20 Development

The upgrade of the KI-TENEX system to the 2060 required a very large effort. Whereas the KI-TENEX system contained a great many local enhancements and adaptations, our goal was to run a TOPS-20 system that was broadly supported but which also tracked research developments outside of those motivated by vendor commercial interests. The most obvious choice for our immediate system peer community was the other 6 DEC 2060 sites at Stanford since we shared common internet problems and also had common goals in supporting research work rather than production computing. We also, of course, retained contact with the other ARPANET computer science systems. This course has constrained our own local developments by being part of a larger group of peers but the added problems of coordination have required fewer site-specific extensions and customizations at the operating system level.

Given this perspective, the following are specific areas of TOPS-20 system effort:

- In the conversion from TENEX, much planning and effort went into moving the file system, along with the pertinent user-specific directory information. In addition, we were able to preserve access to the vast magnetic tape library of archived and otherwise backed up files that had been created and saved since the inception of SUMEX. A TOPS-20 version of BSYS, a file archiving system, was imported from ISI as part of the effort to convert to the 2060. Numerous changes were made to make it compatible with the version of BSYS previously used at SUMEX. The LOOKUP program, used under TENEX, was converted to TOPS-20 use and made compatible with the new version of BSYS. We reviewed and updated appropriate documentation files in the HLP: and DOC: directories. And we identified and upgraded numerous system utility programs that utilized TENEX-dependent system calls.

- Using Tenex code previously developed at SUMEX as a base, we added new code to the TOPS-20 monitor to significantly enhance the user interface to the file system naming primitives. One addition was intercepting a ? typed by a user as part of a file name, then displaying for the user the valid file name alternatives matching the type-in up to that point, and finally returning to the original context, allowing the user to continue typing where he left off. Another addition was to generalize the logic involved in file name recognition in the case where more than one file matches what is typed in at the point where the request for recognition was given. The new logic looks ahead at the alternatives and fills out as much of the file name as possible, i.e. up to the point of ambiguity.

- Continued development of QANAL (formerly ANAL), a crash analysis program that has been under development since 1978. This program significantly eases the burden of analyzing the causes of system crashes due to both hardware and software problems. In addition, the accumulated outputs from QANAL allow for the detection of long term crash correlations to analyze infrequent problems.

- Track network protocol and service (e.g., file transfer and electronic mail) developments. We coordinated SUMEX's changes required to support the ARPANET-wide change from the old NCP protocols to the DOD IP/TCP protocols. This complex software required significant effort on our part because SUMEX-AIM has become a major communications crossroads and so exercises the network code very heavily. This has raised many problems of bugs and performance that we have worked to improve. We have played an active role in network discussion groups related to areas such as electronic mail, network designs, and protocols and had kept system tables for network host names and addresses, both local and over the ARPANET, up-to-date.

- Developed expanded file system support through multiple RP07 disk drive service. We were the first site to support more than one RP07 unit in a single structure.

- Implemented support for the old but superior LP10 printer from the KI-TENEX system. Even though DEC doesn't support this configuration, the LP10 has become our standard printer.

- Implemented subdirectory access to allow users full "owner" access to their subdirectories via the Access Control Job.

- Developed improved system allocation code, including the ability to withhold scheduler "windfall" from a given class or classes, with associated code in SKED% JSYS.

- Improved the efficiency of file backup and archive facilities by flagging directories with ARCHIVE and MIGRATE requests pending rather than searching through all directories serially.

- We have done substantial work on the TOPS-20 system Executive, the program that serves as the primary interface between users and the system. It provides commands to manipulate files, directories, and devices; control job and terminal parameter settings; observe job and system status; and execute public and private programs. The SUMEX EXEC is quite well developed at this stage but we have made several improvements. For example, we added a command line editor developed at the University of Texas and commands for the various laser printer spooling capabilities described later. There were also many more minor upgrades such as reading SYSTEM:LOGIN.CMD and SYSTEM:COMAND.CMD files on user login, account verification, enhancing various information commands, and improved directory and file system facilities to assist users in managing their files.

We have made numerous monitor bug and hardware problem repairs to provide for more reliable system operation and file integrity. Obvious bugs were removed long ago so those remaining are elusive and difficult to track down. We have also spent time keeping up-to-date with the latest monitor releases.

VAX 4.2 BSD UNIX Development

We run UNIX on our shared VAX 11/780 and on our 11/750 file servers. This system has been used pretty much as distributed by the University of California at Berkeley, except for local network support modifications. The local VAX user community is small so we have not expended much system effort beyond staying current with operating system releases and with useful UNIX community developments. The SUMEX VAX was the first site at Stanford to bring up the Berkeley 4.2 BSD distribution in October 1983. Since this was an early distribution, there were quite a number of bug fixes required; these were accomplished both through local effort and through monitoring the unix-wizards mailing list. After this kernel was running on the SUMEX machine, it was transported other sites and became the basis for the campus-wide UNIX 4.2 distribution.

To allow the UNIX network interface code to work in our Stanford subnet environment, we created a pseudo-network interface driver called 'sub0', that routed all output IP datagrams, based on their subnet numbers. This driver was done transparently, so that at system boot time, you could configure the machine for Stanford subnets, or for normal network routing. We also worked with other Stanford sites to install the Stanford PUP network drivers and servers back into 4.2 BSD (Berkeley does not support these).

Workstation System Development

Lisp workstations represent the major new direction for system development at SUMEX-AIM because these machines offer high performance Lisp engines, large address spaces required for sophisticated AI systems, flexible graphics interfaces for users, state-of-the-art program development and debugging tools, and a modularity that promises to be the vehicle for disseminating AI systems into user environments. We have accordingly invested a large part of our system effort in developing selected workstations and the related networking environments for effective use in the SUMEX-AIM community.

Xerox D-Machines

Much of the SUMEX-AIM community uses InterLisp and has moved naturally to the Xerox D-machines -- initially the Dolphin and then the Dandelion, Dandetiger, and Dorado. Much work has gone into hardware installation and networking support but we have also developed numerous software packages to help make the machines more effective for users and to ease our own problems in managing the distributed workstation environment.

In the transition to workstations as *computing* environments suitable for AI applications work, not just as programming environments, much system development remains to be done. One of the problems we have examined and plan to continue to exploring is that of building distributed expert systems. We are interested, for example, in separating the reasoning components and user interfaces and are designing a system with multiple processes which can run on a single or multiple workstations in order to independently develop, tune and evaluate the components. To facilitate this we have developed a prototype inter-process message passing interface which makes the topology of the system invisible to communicating processes, whether on one machine or several CPU's linked via the Ethernet.

Another of our interests is in exploring how to combine different software and/or hardware architectures in order to take advantage of the best features of each. One simple low level program that we built allows us to use Interlisp workstations to down load software into Mesa workstations in order to boot them using the Ethernet as an alternative to the hard or floppy disk drives. Along the same lines, we are exploring efficient ways to communicate high level descriptions of graphic data among differing media. We have developed a simple system which will take text formatting files and translate them into graphic window displays, defining active regions of the screen in the process. This facilitates the design of user interfaces using the familiar medium of text processing.

In our AI systems work, we have developed a low overhead object-oriented system which is designed to be flexible enough to model different object-oriented programming styles at the same time. It is also designed to facilitate a model of large knowledge bases which reside principally on file servers but whose components are loaded on demand. With this system, a minimal set of information about all the objects in a knowledge base is loaded upon opening. This information allows many simple inquires about the nature of objects and their relationships to be made without the main body of the object being resident. Only when non-trivial operations are performed are the contents of the object brought into core. This design is based on the belief that the size of knowledge bases will eventually grow to exceed the capacity of any given computer. However, most systems will generally only need a manageable subset of objects at runtime.

Other work we have done includes monitoring tools to examine static function calling hierarchy as well as view runtime executions graphically. We are also developing graphics interfaces to knowledge base construction and maintenance.

Some of the InterLisp software packages that have been written in the course of this work include:

ACFontCreate -- Reads a Xerox PARC font file in AC format into a lisp data structure

BaudRate -- Benchmarks baudrates by BINing through a file

DSys -- Monitors D machine usage on demand

GraphNet -- Derives topology of the PUP internet via net and gateway probes

HPColor -- Interlisp image stream implementation to drive H-P dgl graphics

Impress -- Interlisp image stream implementation to generate Impress print files

MakeStrike -- Writes out an Interlisp display font as a strike file

MLabel -- Generates mailing labels from a mailing list

RasterFontCreate -- Generates an Impress font of bitmap patches in arbitrary scale

ReadRSTFontFile -- Reads an Impress font file into a list data structure

RemoteTools -- Tools to manipulate a remote Interlisp using its systat process

RootPicture -- Reads a Press file bitmap into a lisp bitmap

RSTSample -- Creates an Impress sampler showing all characters of a font

SIL -- Reads and displays a SIL drawing file and optionally hardcopies it

SYSTAT -- a remote Eval server for Interlisp

Undither -- Compresses a previously dithered image into an AIS file

VDSDog -- Monitors array space usage to prevent crashing from lack thereof

WriteRSTFontFile -- generates an Impress font file from a special Lisp structure

ZDir -- TENEX-style directory lister for use with UNIX via Leaf server calls

DScribe -- A simple SCRIBE-to-display list parser/driver.

EtherBoot -- Provides microcode and program boot service for Xerox 8000's

GraphCalls -- Graphs the calling hierarchy of a lisp function and more

Hash -- Provide a machine independent hash file facility

EditBG -- A background/border texture editor.

FileLstW -- Menu-based interface to the file package.

MagnifyW -- A magnifying glass for bitmaps.

Message -- Multi-process/Multi-CPU message passing facility.

MultiW -- Links windows so that they move, surface, and close as a group

OZone -- An object-oriented programming system for Interlisp

Plotter -- Interlisp image stream to generate native-mode H-P plot files

Register -- Bundles menus into a coherent device for complex input

Region -- A utility to allow dissimilar activity in a single window.

Storage -- A utility to display Interlisp data type storage graphically.

Once a package has been developed and determined to be of general interest, we announce it over an electronic mail users list and make it available to other sites. In some cases, packages have such extensive utility that they are submitted as LispUsers packages for distribution by Xerox. This occurred in the case of Graphcalls, Hash, MultiW, and FileLstW, the latter submitted under the name Manager.

We have worked closely with many other sites, including the Center for Study of Language and Information at Stanford, the Stanford Campus Networking group, Rutgers University, Ohio State University, the University of Pittsburgh, Cornell, Maryland, and industrial research groups such as Xerox Palo Alto Research Center, SRI, Teknowledge, IntelliCorp, and Schlumberger-Doll Research. We have been the maintainers for the international electronic mail network of users for research D-machines, which have upwards of 300 readers, and the interchange of ideas and problems among this group has been of great service to all users.

Symbolics Lisp Machines

We have a growing community of Symbolics machines and users. Little development has gone into the tools for these systems yet because the small number of machines we have are concentrated in applications groups. We have actively supported the installation and maintenance of these systems, the installation of new software releases, and the integration of these systems with the rest of our networking environment. We were a beta test site for the Symbolics IP/TCP software.

Macintosh Workstations

In early 1984 Apple Computer released their new Macintosh and we were immediately interested in it as a possible low-cost display workstation to interface to our Lisp workstations and other hosts. In order to evaluate the Macintosh for this purpose, SUMEX received some early equipment and manuals through Stanford's participation in the Apple university consortium program. Like many groups trying to experiment with Macintosh software however, we found the Apple Lisa cross-development environment somewhat restrictive and hard to use and this was the only way to create Macintosh software at the time. So we built a UNIX-based cross-development environment on our VAX. It turns out, that this was the first C development environment available on the Macintosh when we released our software (via Arpanet FTP) in June of 1984. SUMacC (Stanford University Macintosh C) has been quite widely received, and is in use at well over a hundred sites throughout the US and in foreign countries. SUMACC integrated pieces of software from many groups, and was therefore something of a cooperative effort. We have openly distributed it to other users either through network FTP or a magnetic tape at distribution cost. Version 2.0 of the SUMACC system was released in November of 1984.

Among the many useful programs subsequently written with SUMACC were: (1) a Kermit program done at Harvard, (2) the Mac PSL (Portable Standard LISP) done at the University of Utah, and (3) an 'external file system' done by John Seamons of LucasFilm which allows the Macintosh to use an Ethernet host (such as UNIX) as a general network file server (see also page 95).

With the increased usage of Macintoshes in the SUMEX-AIM community, the need to be able to transfer files between them and TOPS-20 mainframes quickly arose. We therefore reimplemented the MACGet and MACPut file transfer utilities, previously developed for UNIX, for TOPS-20. These incorporated TOPS-20 style terminal handling and file system conventions. Both programs provide reliable (i.e., checksummed) transfer of either text or binary data, and are now gaining wide-spread use outside of SUMEX.

Virtual Workstation Graphics

Finally, we have done a number of experiments with the remote connection of bitmapped displays to hosts and workstations. Generally, the displays on Lisp machines are tethered through a high bandwidth cable to their processors. This limits the flexibility with which users can move from one Lisp machine to another (one must move physically to another machine) and loses the ability of researchers to work from home over telephone lines. A way of providing more flexible display to processor connection is to use a virtual graphics protocol, such as the V Kernel system developed by Lantz [37], that allows efficient communication of the contents to be displayed on a bitmapped screen. In an initial experiment, an Interlisp virtual graphics module was written to run on the DEC-2060 and drive the graphics engine of a Sun MicroSystems workstation over the Ethernet. This system allows applications running on the DEC-2060 to create views, and windows within those views on the remote workstation, and then using the Virtual Graphics Terminal Protocols, manipulate those views and windows. One can place text, draw objects such as points, lines, shaded rectangles, splines, and bitmaps in these screen areas. Local and remote editing of the graphics representation is also possible with a responsiveness close to that of a directly connected display.

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.

Remote Networks

We continue our connection to TYMNET as the primary means for access to SUMEX-AIM from research groups around the country and abroad. Substantial work was required to transfer TYMNET service from the KI-TENEX system to the 2060 because the new system does not support the same memory-sharing interface we had for the KI-10's. There has been no significant change in user service or network performance though. Very limited facilities for file transfer exist and no improvements appear to be forthcoming soon. Services continue to be purchased jointly with the Rutgers Computers in Biomedicine resource to maximize our volume usage price break. We continue to have serious difficulties getting needed service from TYMNET for debugging network problems and users away from major cities have problems with echo response times.

We also continue our extremely advantageous connection to the Department of Defense's ARPANET, managed by the Defense Communications Agency (DCA). 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. Terminal access restrictions are in force so that only users affiliated with DoD-supported contractors may use TELNET facilities. ARPANET is the primary link between SUMEX and other machine resources such as Rutgers-AIM and the large AI computer science community supported by DARPA. Our early Honeywell IMP has been upgraded to a BBN C/30 IMP in preparation for the transition to the IP/TCP protocols. We are also investigating the installation of a link to the DARPA wideband satellite network to facilitate the rapid transfer of large amounts of data such as are involved with projects like our Concurrent Symbolic Computing Architectures project.

Local Area Networks

For many years now, we have been developing our 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 and are now the standard system used in the campus-wide Stanford University network. We are developing gateways to interface other equipment as needed too (e.g., the Macintosh and Lisa). A diagram of our local area network system is shown in Figure 5 on 94 and the following summarizes our LAN-related development work.

MC-68000 Server Kernel -- Our early network gateways and TIP's were based on PDP-11 systems. But these soon became limiting in terms of speed, address space, and cost. With the introduction of the Motorola MC-68000 microprocessor and its integration into a compact, large-memory machine in the prototype SUN processor board developed in the Computer Systems Laboratory at Stanford, a much better vehicle was at hand. The net server software we developed for the PDP-11 included a kernel which handles hardware interfaces, core allocation, process scheduling, and low-level network protocol management. The 3 MBit/sec Ethernet PDP-11 based PUP kernel was translated and augmented for the MC-68000 CPU/SUN ethernet interface. This kernel then became the basis for the SUMEX gateway and TIP software which both have become the Stanford standard. As networking technology developed, the SUMEX kernel was extended to include 10 MBit/sec Ethernet drivers and to support 10 Mbit/sec PUP, XNS, and IP protocols. The main modification needed was the addition of a 10 MBit/sec Ethernet address resolution protocol module so that a 10 MBit/sec PUP host could discover its "soft" PUP address from a cooperating gateway on its local network.

Ethernet TIP -- Based on the new augmented MC-68000 kernel, the 3 Mbit/sec PDP-11 Ether TIP code was translated. This new TIP could handle increments of 8

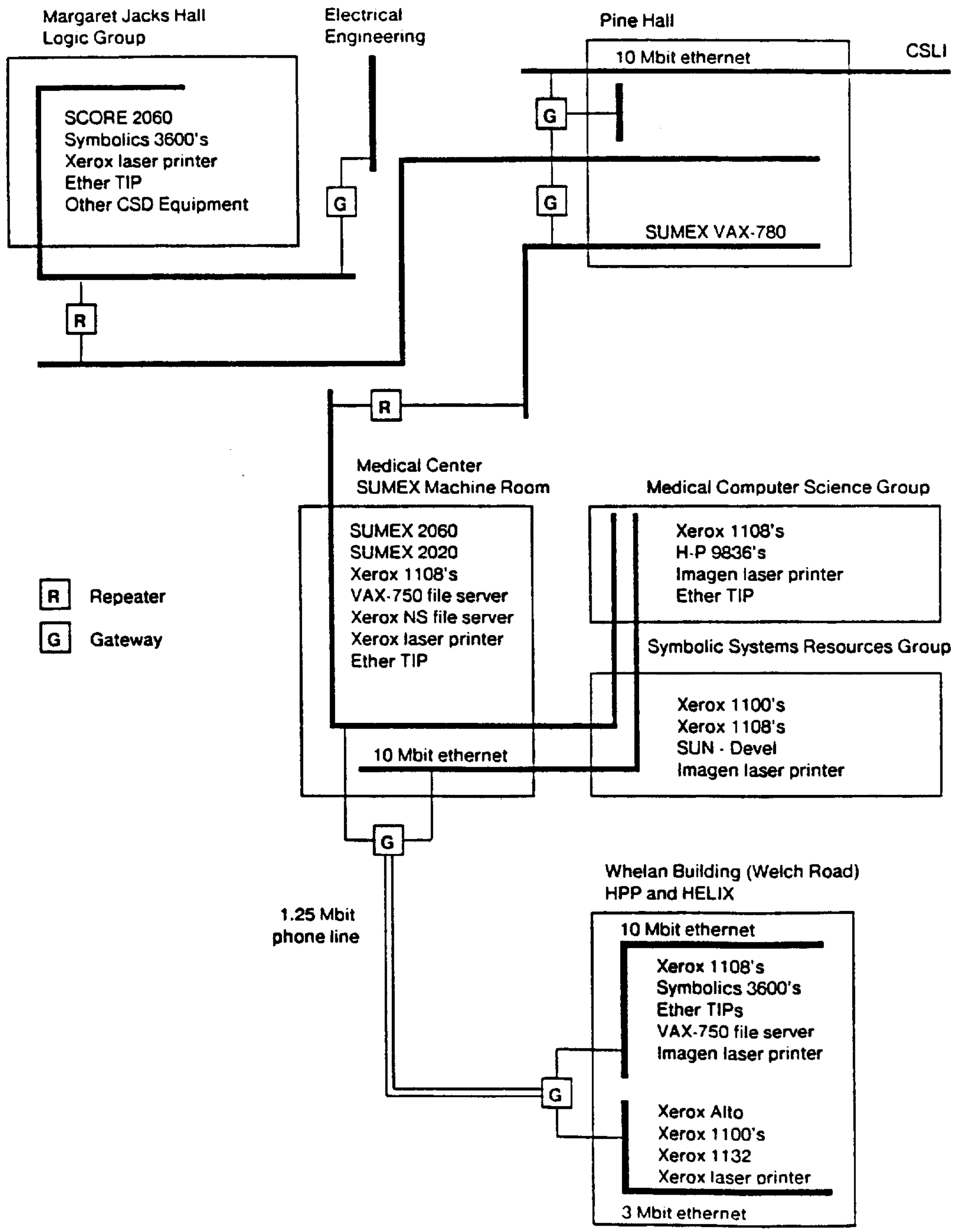

Figure 5: SUMEX-AIM EtherNet Configuration

lines up to 32 lines in a six slot backplane. With the advent of the newer 16 line DUART's developed in the Stanford Computer Science Department, 80 line TIP's have been built using this TIP code. This code is still running on several 3 Mbit/sec Ether TIP's at SUMEX. As 10 Mbit/sec networks were introduced, the TIP code was updated and adapted so that TIP's could run on either 3 MBit/sec or 10 MBit/sec Ethernets. There are now over 20 TIP's installed at Stanford using the SUMEX code and the number will increase substantially as the campus-wide local area network grows. The development of this software is essentially complete now with the recent addition of an improved user interface and facilities for inbound connections (such as for remote printers).

Ethernet Gateways -- Like the TIP systems, the PDP-11 gateway code was adapted to the MC-68000 hardware and extended to both 3 Mbit/sec and 10 Mbit/sec networks. Gateways can be configured to support up to four directly connected networks which may be either 3 MBit/sec or 10 MBit/sec. The gateway system was made "self-configuring" so that only one bootable gateway was needed. Network directory downloading and name/address lookup services were added. The routing algorithm was rewritten to minimize probe time for efficiency because of the continued growth of the number of subnetworks in the Stanford University network. The gateway now supports PUP and IP packet transport and XNS packet routing for both 10mb and 3mb networks is being completed. There are over twenty SUMEX gateways installed at Stanford and this number should double in the next year.

A special gateway configuration was required for the HPP move to Welch Road. Since the physical link was differentially driven 1.25 MBit/sec twisted pair cable, the network connections required two three-way gateways, one at either end, and special hardware to interface the serial lines with the ethernet interfaces. The required special hardware and software were built and the WR gateway has operated very effectively.

Apple Gateway Another special gateway, named SEAGATE, was developed to better integrate the Apple Macintosh into our Ethernet system. It links the Ethernet and Apple's AppleBus/AppleTalk network. This was completed and released in February 1985. Several internet sites, including some at Stanford, are currently constructing duplicate gateways. Also, several commercial firms are building a one board version of the gateway which should lower the cost to about $1000 per gateway. EFS, MAT, and AppleTalk Library are some sample Macintosh programs and UNIX daemons, that utilize SEAGATE. EFS is an external file system, written by John Seamons, and modified by us to work over AppleTalk. With EFS the Mac user sees his normal iconic view of the world. His UNIX directory appears as an icon and he can remotely execute and transfer files, simply by clicking on their icons. EFS is to the Mac as Leaf is to a LISP machine. The AppleTalk library is used by all of these programs to perform the ATP protocol (AppleTalk transaction protocol). This is the general protocol used to perform printing, file transfer, etc. with the Mac. The library allows a UNIX user-level process to perform this ATP protocol. Note that no kernel changes are required, since the ATP datagrams are imbedded in IP datagrams (UDP) by the SEAGATE. MAT is the Mac ATP Transfer program, a sample program that does file transfers with a UNIX host. It can also act as the framework for a Mac mail or print service.

Remote File Service -- In a distributed workstation environment, effective file access and transfer facilities between workstations and other hosts and servers are a must, especially to file servers like those we built around VAX 11/750 UNIX systems. Initial file service support used code written as a student project in the Stanford Computer Systems Laboratory. But as the number of workstations increased, service degraded and it became necessary to rewrite the PUP/BSP UNIX software package, and major portions of those programs dependent upon these protocols. This resulted in a 300% increase in throughput and stabilized the Lisp Machine to VAX 11/750 file service environment. At the same time we made major improvements to the UNIX Leaf service for XEROX D-machines. The earlier code, again a student systems project, had many bugs and inefficiencies and required a complete rewrite. In the new code, each Leaf connection was given a separate process to manage its Leaf resources, whereas previously, all users' Leaf requests were simply handled as a serial queue. This meant that every packet created a bottleneck for its successors. This work resulted in a much better Leaf service environment with considerable improvement in overall responsiveness and throughput.

Laser Printing Services

Since the first Xerox laser printers were developed in the mid-1970's, several companies have produced computer-driven systems, such as the Xerox Raven and the Imagen 8/300. These systems have become essential components of the work 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.

We collaborated to develop an Ethernet interface for Imagen printers starting about January of 1984. We arranged to upgrade our Imprint-10 controller in exchange for the UNIX software needed to drive it from the network and were the first site to receive this controller in beta test stage. The UNIX software we developed made it possible to connect the printer to the new 4.2 BSD line printer spooler package using IP/TCP protocols. This was completed about March of 1984. After the UNIX implementation was complete, we developed the corresponding TOPS20 software to interface to this new printer and later, integrated it into the TOPS20 Galaxy spooler package. Other sites on campus and in the internet, began using the new printer and our spooling software as well.

We similarly developed and enhance the spooling system for the Dover and Alto-Raven laser printers and added a header page for Raven output to separate listings. And in addition to the device support for the printers to interface to the various mainframe hosts machines in our network, we also developed packages to allow Xerox D-machines and Symbolics 3600 machines to print to the networked laser printers.

On the SUMEX-AIM mainframe hosts, SCRIBE is the predominant document compilation system, but in the initial stages, it was essentially only used with the Xerox Dover printer or a daisywheel typewriter. In the succeeding years we integrated the Imagen Imprint-10 driver from Unilogic, brought up the Xerox Alto-Raven, and installed support for the new group of Imagen printers (the 8/300's), which are based on a Canon copier and are now the workhorse printing resources of the local community. We made numerous improvements in the printing fonts available to users, including a rework of Knuth's Computer Modern Roman fonts for a more contemporary look on the Imprint-10, creating a sans serif font family based on Computer Modern Roman, generating Helvetica and Times Roman font families from the Xerox sources used to generate the Dover fonts, and creating and improving many document types in use by the community.

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, DEC-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 on operating systems and networking. A further gratifying example is the TTYFTP program, originally written at SUMEX as a system for file transfers usable over any circuit that appears as a terminal line to the operating system (hardline, dial-up, TYMNET, etc.) and incorporating appropriate control protocols and error checking. The design was derived from the DIALNET protocols developed at the Stanford AI Laboratory with extensions to allow both user and server modules to run as user processes without operating system changes. TTYFTP formed the basis for the KERMIT program that is now distributed by Columbia University and which is in very wide use for communications between personal computers and to mainframe hosts.

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 ARPANET 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 109).

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. Of course, the upgrade of the KI-TENEX system to the 2060 required major planning and care to ensure continuous resource operation during the phase-over. Similarly, the relocation of our VAX 11/780 to Pine Hall and the outfitting of the KSL machine room at the Welch Road laboratory required much effort.

We use students for much of our operations and related systems programming work. Over the past 4 years, we have hired and trained a total of 15 undergraduate operations assistants.

We also 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.

2.1.4.5. Core AI Research

We have maintained a strong core AI research effort in the SUMEX-AIM resource aimed at developing information resources, basic AI research, and tools of general interest to the SUMEX-AIM community. It should be noted that the SUMEX resource grant from NIH provides much of the computing environment for this core AI work[1] but NIH supports only a small part of the manpower and other support for core AI. For example, NIH has provided partial funding for work on the AI Handbook, the AGE project, and part of the core ONCOCIN development for the dissemination of consultative AI systems. Substantial additional support for the personnel costs of our core AI research (roughly comparable to the NIH investment in computing resources) comes from DARPA, ONR, NSF, NASA, and several industrial basic research contracts to the Knowledge Systems Laboratory or KSL[2] (see the summary of core research funding on page 105).

Our core AI research work has long been the mainstay on which our extensive list of applications projects are based. This work has been focused on medical and biological problems for over a decade with considerable success, particularly in the area of expert systems which represent one important class of applications of AI to complex problems -- in medicine, science, engineering, and elsewhere. Numerous high-performance, expert systems have resulted from our work on expert systems in such diverse fields as analytical chemistry, medical diagnosis, cancer chemotherapy management, VLSI design, machine fault diagnosis, and molecular biology. Other projects have developed generalized software tools for representing and utilizing knowledge (e.g., EMYCIN [6, 68], UNITS [66], AGE [54], MRS [20], GLISP [57]) as well as comprehensive publications such as the three-volume *Handbook of Artificial Intelligence* [1] and books summarizing lessons learned in the DENDRAL [43] and MYCIN [6, 65] research projects.

But the current ideas fall short in many ways, necessitating extensive further basic research efforts. Our core research goals 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.

The following summary reports progress on the basic or core research activities within the KSL. As indicated earlier, the development of the ONCOCIN system (under Professor Shortliffe) is an important part of our core research proposal for the renewal period. Progress on that work is reported separately in Section 6.1.3 on page 209, however, because its efforts have been supported as a collaborative and resource-related research project up until now. Together, this work explores a broad range of basic research ideas in many application settings, all of which contributes in the long term to improved knowledge based systems in biomedicine.

Recent Highlights of Research Progress

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 Hodgkins disease. Thus, within the KSL

---

[1] DARPA funds have also helped substantially in upgrading the KI-TENEX system to the 2060 and in the purchase of community Lisp workstations

[2] See Appendix A on page 285 for an overview of the KSL organization.

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.

In addition to the technical reports listed later, the following books and survey articles were published just during this year -- 11 books total have been published in the past 4 years as indicated in Appendix A. These are of central interest to AI researchers and of direct relevance to the mission of the SUMEX-AIM resource.

BOOKS:

1. Buchanan, B.G. and Shortliffe, E.H., eds. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley Publishing Company, 1984.

2. Clancey, W.J. and Shortliffe, E.H., eds. *Readings in Medical Artificial Intelligence: The First Decade.* Reading, MA: Addison-Wesley Publishing Company, 1984.

3. Cohen, Paul R. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach.* London and Marshfield, MA: Pitman Advanced Publishing Program, 1985.

SURVEY ARTICLES: HPP 84-15, 84-20, 84-23, 84-28, and 84-32.

In addition, work is progressing on a textbook for students beginning to study medical computing and artificial intelligence[1]. This multi-authored volume should be completed in draft form by the end of 1985 and a 1986 publication date is contemplated. Writing this new book will be facilitated by the SUMEX resource, much as the *Handbook of AI* was in the past. A multi-authored text of this type, particularly one for which the authors are spread at numerous different universities around the country, would be a nightmare to compile if it were not for the SUMEX resource. Many of the contributors to the book have been assigned SUMEX accounts for purposes of manuscript preparation. On-line manuscript work through the shared facility, coupled with messaging capabilities, will greatly enhance the efficiency and accuracy of the developing chapters and the editing process.

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

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?

   A working version of NEOMYCIN has been implemented which demonstrates the effectiveness of representing strategy knowledge explicitly. A detailed study of rule-based systems was published in book form. Specific representational issues in logic-based systems were addressed in the

[1]Shortliffe, E.H., Wiederhold, G.C.M., and Fagan, L.M.; *An Introduction to Medical Computer Science*, Reading, MA: Addison-Wesley (in preparation).

context of MRS. We designed a method for representing temporal knowledge in ONCOCIN. Finally, Cooper's Ph.D. thesis on representing and using causal and probabilistic knowledge was published in this year.

[See KSL technical memos KSL-84-9, KSL-84-10, KSL-84-18, KSL 84-31, KSL-84-41, KSL-85-5.]

2. *Advanced Architectures and Control*: What kinds of software tools and system architectures can be constructed to make it easier to implement expert programs with increasing complexity and high performance? How can we design flexible control structures for powerful problem solving programs?

   Much of our research in the past year has involved investigations with the Blackboard architecture begun in previous years. We have implemented our design in a working system called BB1.

   [See KSL technical memos KSL-84-11, KSL-84-12, KSL-84-14, KSL 84-16, KSL 84-36.]

3. *Knowledge Acquisition*: How is knowledge acquired most efficiently -- from human experts, from observed data, from experience, and from discovery? How can a program discover inconsistencies and incompleteness in its knowledge base? How can the knowledge base be augmented without perturbing the established knowledge base?

   Three Ph.D. theses (Fu, Greiner, and Dietterich) in the area of knowledge acquisition were completed in this year. Fu's work develops methods for learning by induction, where the target rules may have some associated degrees of uncertainty and may contain names of intermediate concepts. This work was demonstrated in the context of diagnosing causes of jaundice. Greiner's work examines learning by analogy. Dietterich's work elucidates methods needed in learning programs to deal with state variables and with problems of using a partially learned theory to interpret new data that will be used to learn new elements of the theory. In addition, we implemented the first parts of a program that can learn by watching an expert. And we implemented a prototype system that learns control heuristics from an expert using a problem solving program written in BB1.

   [Preliminary results have been published in KSL-84-10, KSL-84-18, KSL-84-24, KSL-84-38, KSL-84-45, KSL 84-46, KSL-85-2, KSL-85-4.]

4. *Knowledge Utilization*: 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?

   We completed the design of a system using Dempster's rule of propagating uncertainty, and we examined several other issues regarding the use of probabilistic information in expert systems. Dr. Jean Gordon, a mathematician and Stanford medical student, collaborated with Dr. Shortliffe on work that examines inexact inference using the Dempster-Shafer theory of evidence, demonstrating its relevance to a familiar expert system domain, namely the bacterial organism identification problem that lies at the heart of the MYCIN system, and presenting a new adaptation of the D-S approach with both computational efficiency and permitting the management of evidential reasoning within an abstraction hierarchy.

   We examined the use of counter-factual conditionals in logic-based systems and completed an analysis of how procedural hints can be used by a problem solver.

[See KSL technical memos KSL-84-11, KSL-84-17, KSL-84-21, KSL-84-30, KSL-84-31, KSL-84-35, KSL 84-41, KSL-84-42, KSL-84-42, KSL-84-43.]

5. *Software Tools*: How can specific programs that solve specific problems be generalized to more widely useful tools to aid in the development of other programs of the same class?

   We have continued the development of new software tools for expert system construction and the distribution of packages that are reliable enough and documented so that other laboratories can use them. These include the old rule-based EMYCIN system, MRS, and AGE. Progress has been made in making the BB1 instantiation of the blackboard architecture domain-independent. We have begun constructing and editing subsystems and have completed a first implementation of an explanation subsystem.

   [See KSL technical memos KSL-84-16, KSL-84-39.]

6. *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? How can knowledge in a system be transferred effectively to students and trainees?

   A program for inferring a model of users was designed and implemented in the context of a tutoring system that aids in teaching algebra. A second user-modelling program was implemented in the context of NEOMYCIN to help understand how an expert solves problems. A survey of explanation capabilities in medical consultation programs was published.

   A new project on knowledge-based explanations in a decision analysis environment is getting underway as the thesis research of Dr. Glenn Rennels. This work is actually a synthesis of artificial intelligence, decision analysis and statistics. The work concerns medical management, not diagnosis; diagnostic decisions identify underlying mechanisms of the illness, and group the patient's problems under a diagnostic label, whereas management decisions plan actions that will prevent undesirable outcomes and restore health. The intelligent behavior we want to emulate is (a) the identification of studies relevant to a given clinical case, and (b) interpretation of those studies for decision-making assistance.

   [See KSL technical memos KSL-84-12, KSL 84-27, KSL-84-29.]

7. *Planning and Design*: What are reasonable and effective methods for planning and design? How can symbolic knowledge be coupled with numerical constraints? How are constraints propagated in design problems?

   A major paper on skeletal planning was published in this year. And we published in the biochemistry literature some results of applying skeletal planning to experiment design in genetic engineering.

   [See KSL technical memos KSL-84-33, KSL-85-6.]

8. *Diagnosis*: How can we build a diagnostic system that reflects any of several diagnostic strategies? How can we use knowledge at different levels of abstraction in the diagnostic process?

   Research on using causal models in a medical decision support system (NESTOR) was published in this year. Using the domain of hypercalcemic disorders, NESTOR attempts to use knowledge-based methods within a formal probability theory framework. The system is able to score hypotheses with causal knowledge guiding the application of sparse probabilistic knowledge; search for the most likely hypothesis without exploring the entire hypothesis space; and critique and compare hypotheses which are generated by the system, volunteered by the user, or both.

A second medical diagnosis program that uses causal models of renal physiology (AI/MM) was also published. In this system, analysis and explanation of physiological function is based on two kinds of causal relations: empirical "Type-1" relations based on definitions or on repeated observation and mathematical "Type-2" relations that have a basis in physical law. Inference rules are proposed for making valid qualitative causal arguments with both kinds of causal basis.

A working implementation of the PATHFINDER system was evaluated and its diagnostic strategies were analyzed. A taxonomy of diagnostic methods was completed and integrated into the NEOMYCIN system.

[See KSL technical reports: KSL-84-13, KSL-84-19, KSL-84-48, KSL-85-5.]

Relevant Core Research Publications

HPP 84-9 David H. Hickam, Edward H. Shortliffe, Miriam B. Bischoff, A. Carlisle Scott, and Charlotte D. Jacobs; *Evaluations of the ONCOCIN System: A Computer-Based Treatment Consultant for Clinical Oncology, (1) The Quality of Computer-Generated Advice and (2) Improvements in the Quality of Data Management,* May 1984.

HPP 84-10 Thomas G. Dietterich; *Learning About Systems That Contain State Variables,* June 1984. In *Proceedings of AAAI-84,* August 1984.

HPP 84-11 M. Genesereth, and D.E. Smith; *Procedural Hints in the Control of Reasoning,* May 1984.

HPP 84-12 Derek H. Sleeman; *UMFE: A User Modelling Front End Subsystem,* April 1984.

HPP 84-13 Eric J. Horvitz, David E. Heckerman, Bharat N. Nathwani, and Lawrence M. Fagan; *Diagnostic Strategies in the Hypothesis-Directed PATHFINDER System,* June 1984, submitted to the *First Conference on Artificial Intelligence Applications, Denver, CO.,* December 5-7, 1984.

HPP 84-14 Vineet Singh, and M. Genesereth; *A Variable Supply Model for Distributing Deductions,* May 1984.

HPP 84-15 Bruce G. Buchanan; *Expert Systems,* July 1984, *Journal of Automated Reasoning, Vol. 1, No. 1, Fall,* 1984.

HPP 84-16 *STAN-CS-84-1034* Barbara Hayes-Roth; *BB-1: An Architecture for Blackboard Systems That Control, Explain, and Learn About Their Own Behavior,* December 1984.

HPP 84-17 M.L. Ginsberg; *Analyzing Incomplete Information,* 1984.

HPP 84-18 William J. Clancey; *Knowledge Acquisition for Classification Expert Systems,* July 1984, *Proceedings of ACM-84,* 1984.

HPP 84-19 E.H. Shortliffe; *Coming to Terms With the Computer,* to appear in S.R. Reiser, and M. Anbar (eds.), *The Machine at the Bedside: Strategies for Using Technology in Patient Care, Cambridge University Press,* 1984.

HPP 84-20	E.H. Shortliffe; *Artificial Intelligence and the Future of Medical Computing*, in *Proceedings of a Symposium on Computers in Medicine, annual meeting of the California Medical Association, Anaheim, CA.*, February 1984.
HPP 84-21	E.H. Shortliffe; *Reasoning Methods in Medical Consultation Systems: Artificial Intelligence Approaches (Tutorial)*, in *Computer Programs in Biomedicine* January 1984.
HPP 84-22	*ONCOCIN Project: Studies to Evaluate the ONCOCIN System; 6 Abstracts*, February 1984.
HPP 84-23	Edward H. Shortliffe; *Feature Interview: On the MYCIN Expert System*, in *Computer Compacts, 1:283-289*, December 1983/January 1984.
HPP 84-24	B.G. Buchanan, and E.H. Shortliffe; *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, published with *Addison-Wesley, Reading, MA.*, 1984.
HPP 84-25	W.J. Clancey, and E.H. Shortliffe; *Readings in Medical Artificial Intelligence: The First Decade*, published with *Addison-Wesley, Reading, MA.*, 1984.
HPP 84-27	Edward H. Shortliffe; *Explanation Capabilities for Medical Consultation Systems (Tutorial)*, in D. Lindberg, and M. Collen (eds.), *Proceedings of AAMSI Congress 84, pp. 193-197, San Francisco*, May 21-23, 1984.
HPP 84-28	E.H. Shortliffe, and L.M. Fagan; *Artificial Intelligence: The Expert Systems Approach to Medical Consultation*, in *Proceedings of the 6th Annual International Symposium on Computers in Critical Care and Pulmonary Medicine, Heidelberg, Germany*, June 4-7, 1984.
HPP 84-29	David C. Wilkins, Bruce G. Buchanan, and William J. Clancey: *Inferring an Expert's Reasoning by Watching, Proceedings of the 1984 Conference on Intelligent Systems and Machines*, 1984.
HPP 84-30	M.L. Ginsberg: *Non-Monotonic Reasoning Using Dempster's Rule*, June 1984.
HPP 84-31	M.L. Ginsberg: *Implementing Probabilistic Reasoning*, June 1984.
HPP 84-32	Bruce G. Buchanan: *Artificial Intelligence: Toward Machines That Think*, July 1984, in *Yearbook of Science and the Future, pp. 96-112, Encyclopedia Britannica, Inc., Chicago*, 1985.
HPP 84-33	Rene Bach, Yumi Iwasaki, and Peter Friedland; *Intelligent Computational Assistance for Experiment Design*, in *Nuclear Acids Research*, January 1984.
MCS Thesis	Kunz, John C.; *Use of Artificial Intelligence and Simple Mathematics to Analyze a Physiological Model*, Doctoral dissertation, Medical Information Sciences, June 1984.
HPP 84-35	Jean Gordon, and Edward Shortliffe; *A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space*, September 1984 and in *Artificial Intelligence*, 26(3), July 1985.
HPP 84-36	Michael R. Genesereth, Matt Ginsberg, and Jeff S. Rosenschein; *Cooperation Without Communication*, September 1984.

HPP 84-38 Li-Min Fu, and Bruce G. Buchanan; *Enhancing Performance of Expert Systems by Automated Discovery of Meta-Rules*, September 6, 1984.

HPP 84-39 Paul S. Rosenbloom, John E. Laird, John McDermott, Allen Newell, and Edmund Orciuch; *RI-Soar: An Experiment in Knowledge-Intensive Programming in a Problem-Solving Architecture*, to appear in the *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, October 1984.

HPP 84-41 *STAN-CS-84-1032* Michael R. Genesereth, Matthew L. Ginsberg, and Jeffrey S. Rosenschein; *Solving the Prisoner's Dilemma*, November 1984.

HPP 84-42 Matthew L. Ginsberg; *Does Probability Have a Place in Non-Monotonic Reasoning?* submitted to the *IJCAI-85*, November 1984.

HPP 84-43 *STAN-CS-84-1029* Matthew L. Ginsberg; *Counterfactuals*, submitted to the *IJCAI-85*, December 1984.

HPP 84-45 Devika Subramanian, and Michael R. Genesereth; *Experiment Generation with Version Spaces*, December 1984.

HPP 84-46 Thomas G. Dietterich; *Constraint Propagation Techniques for Theory-Driven Data Interpretation*, PhD Thesis, to be published as a book by *Kluwer*, December 1984.

HPP 84-48 *STAN-CS-84-1031* Gregory F. Cooper; *NESTOR: A Computer-Based Medical Diagnostic Aid That Integrates Causal and Probabilistic Knowledge*, PhD Thesis, December 20, 1984.

KSL 85-2 *STAN-CS-85-1036* Barbara Hayes-Roth, and Michael Hewett; *Learning Control Heuristics in BB1*, submitted to the *IJCAI-85*, January 1985.

KSL 85-4 (Needs Authors Permission) Li-Min Fu, and Bruce G. Buchanan; *Learning Intermediate Knowledge in Constructing a Hierarchical Knowledge Base*, submitted to the *IJCAI Conference Proceedings for 1985*, January 1985.

KSL 85-5 (Needs Authors Permission) William J. Clancey; *Heuristic Classification*, March 1985.

KSL 85-6 Peter E. Friedland, and Yumi Iwasaki; *The Concept and Implementation of Skeletal Plans*, published in the *Journal of Automated Reasoning*, 1985.

KSL 85-7 Rene Bach, Yumi Iwasaki, and Peter Friedland; *Intelligent Computational Assistance for Experiment Design*, published in *Nucleic Acids Research*, 1985.

KSL 85-8 (Needs Authors Permission) M.G. Kahn, J. Ferguson, E.H. Shortliffe, and L. Fagan; *An Approach for Structuring Temporal Information in the ONCOCIN System*, March 1985.

Summary of Core Research Funding Support

We are pursuing a broad core research program on basic AI research issues with support from not only SUMEX but also DARPA, NASA, NSF, and ONR. SUMEX provides some salary support for staff and students involved in core research and invaluable computing support for most of these efforts. Additional salary support comes from the sources shown starting on page 36.

Interactions with the SUMEX-AIM Resource

Our interactions with the SUMEX-AIM resource involve the facilities -- both hardware and software -- and the staff -- both technical and administrative. Taken together as a whole resource, they constitute an essential part of the research structure for the KSL. Many of the grants and contracts from other agencies have been awarded partly because of the cost-effectiveness of AI research in the KSL due to the fact that much of our computing needs could be more than adequately met by the SUMEX-AIM resource. In this way the complementary funding of this work by the NIH and other agencies provides a high leverage for incremental investment in AI research at the SUMEX-AIM resource.

We rely on the central SUMEX facility as a focal point for all the research within the KSL, not only for much of our computing, but for communications and links to our many collaborators as well. As a common communications medium alone, it has significantly enhanced the nature of our work and the reach of our collaborations. The existence of the central time-shared facility has allowed us to explore new ideas at very small incremental cost.

As SUMEX and the KSL acquire a diversity of hardware, including LISP workstations and smaller personal computers, we rely more and more heavily on the SUMEX staff for integration of these new resources into the local network system. The staff has been extremely helpful and effective in dealing with the myriad of complex technical issues and leading us competently into this world of decentralized, diversified computing. At the same time, the staff has provided a stable, efficient central time-shared machine running software that has been developed at many sites over many years. Without the dedication of the SUMEX staff, the KSL would not be at the forefront of AI research.

2.1.4.6. Dissemination Activities

Throughout the history of the SUMEX-AIM resource, we have made extensive efforts at disseminating the AI technology developed here. This has taken the form of many publications -- over 45 combined books and papers are published per year from 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 GENET community (DNA sequence analysis software) and the ONCOCIN resource-related research project (see 209).

Books and Publications

A sampling of the recent research paper publications of the KSL was given in the previous section on core AI research progress. The following lists the major books published in the past 4 years from the KSL:

- *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.

Software Distribution

We have widely distributed both our system software and our AI tool software. We have no accurate records of the extent of distribution of the system codes because their distribution is not centralized and controlled. The recent programs such as the TOPS-20 file recognition enhancements, the Ethernet gateway and TIP programs, the SEAGATE AppleBus to Ethernet gateway, the PUP Leaf server, the SUMACC development system for Macintosh workstations, and our Lisp workstation programs are well-distributed throughout the ARPANET community and beyond.

We do have reasonably accurate records of the distribution of our AI tool software because the recipient community is more directly coupled to us and the distribution is centralized:

GENET	Prior to the establishment of the BIONET resource at IntelliCorp, we distributed 21 copies of the DNA sequence analysis programs and databases for both DEC-10 and DEC-20 systems.
EMYCIN	A total of 56 sites have received the EMYCIN [6, 68] package for backward-chained, rule-based AI systems.
AGE	The AGE [54] blackboard framework system has been sent out to 35 sites in versions for several machines.
MRS	The MRS [20] logic-based system for meta-level representation and reasoning has been provided to 76 sites.
Other Programs	Smaller numbers of copies of programs such as the SACON [2] knowledge base for EMYCIN, the GLISP [57] system (now distributed by Gordon Novak at the University of Texas), and the new BB1 [28, 27] system have been distributed.

A number of other software packages have been licensed or otherwise made available for commercial development including DENDRAL (Molecular Designs), MAINSAIL (Xidak), UNITS (IntelliCorp), and EMYCIN (Teknowledge and Texas Instruments).

Video Tapes and Films

The KSL and the ONCOCIN project have prepared several 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. The three tapes include:

- *Knowledge Engineering in the Heuristic Programming Project* -- This 20-minute film/tape illustrates key ideas in knowledge-based system design and implementation, using examples from ONCOCIN, PROTEAN, and knowledge-based VLSI design systems. It describes the research environment of the KSL and lays out the methodologies of our work and the long term research goals that guide it.

- *ONCOCIN Overview* -- This is a 30-minute tape providing an overview of the ONCOCIN project. It gives an historical context for the work, discusses the clinical problem and the setting in which the prototype system is being used, and outlines the plans for transferring the system to run on single-user workstations. Brief illustrations of the graphics capabilities of ONCOCIN on a Lisp workstation are also provided.

- *ONCOCIN Demonstration* -- This 1-hour tape provides detailed examples of the key components of the ONCOCIN system. It begins with a demonstration of the prototype system's performance on a time-shared mainframe computer and then shows each of the elements involved in transferring the system to Lisp workstations.

The GENET Dissemination Experiment

Beginning in early 1980, the MOLGEN project investigators at Stanford have made a new set of computing tools available to a national community of molecular biologists through a guest facility called GENET on the SUMEX-AIM resource. This experimental subcommunity was started to broaden MOLGEN's base of scientist collaborators at institutions other than Stanford and to explore the idea of a SUMEX-like resource to disseminate sophisticated software tools to a generally computer-naive community. The enthusiastic response to the very limited announcement of this facility eventually necessitated SUMEX placing severe restrictions on the scope of services provided to this community.

Three main programs were offered to assist molecular genetics users: SEQ, a DNA-RNA sequence analysis program; MAP, a program that assists in the construction of restriction maps from restriction enzyme digest data; and MAPPER, a simplified and somewhat more efficient version of the MOLGEN MAP program, written and maintained by William Pearson of Johns Hopkins University. Some of the other, more-sophisticated programs being developed through MOLGEN research efforts were not yet available for novice users. However, GENET users had access to the SUMEX-AIM programs for electronic messaging, text-editing, file-searching, etc.

The GENET experiment proved so successful that eventually that community was the single biggest consumer of processor cycles on SUMEX. This overload diverted our very limited computing resources away from our mainline goal of supporting projects developing new AI systems in the medical and biological sciences, including molecular biology. Efforts to secure funds to increase SUMEX capacity for the burgeoning GENET use failed. Thus, without any fair way to allocate a small resource to the growing GENET community and in order to restore the necessary emphasis on biomedical computer science research on SUMEX, it was necessary to phase out the GENET usage. We closed the GENET account at the end of 1982, with a mandate from an ad hoc GENET Executive Committee, and phased out all usage by spring of 1983. In the process, we developed procedures by which academic users could obtain their own copies of the GENET programs used at SUMEX and we provided a list of alternate sources for GENET-like computing services. As indicated above, SUMEX has supplied 21 systems to academic users with compatible machines.

Since the phase-out of GENET at SUMEX, IntelliCorp, a commercial AI company, submitted a proposal to the NIH Division of Research Resources for a BIONET resource and was successful in obtaining funding. The BIONET resource began operation in the summer of 1984.

2.1.4.7. Training Activities

The SUMEX resource exists to facilitate biomedical artificial intelligence applications from program development through testing in the target research communities. 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 gain access to the system and use it effectively. 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.

On another front, 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.

And finally, the training of graduate students is an essential part of the research and educational activities of the KSL. Currently 41 students are working with our projects centered in Computer Science and another 20 students are working with the Medical Computer Science program in Medicine. Of the 41 working in Computer Science, 25 are working toward Ph.D. degrees, and 16 are working toward M.S. degrees. A number of students are pursuing interdisciplinary programs and come from the Departments of Engineering, Mathematics, Education, and Medicine.

Based on the SUMEX-AIM community environment, we have initiated two unique and 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, who are knowledgeable about knowledge-based system techniques.

The *Medical Information Sciences (MIS)* program is 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. The MIS training program is based in School of Medicine, directed by Dr. Shortliffe, co-directed by Dr. Fagan, and overseen by a group of nine University faculty that includes several faculty from the Knowledge Systems Laboratory (Profs. Shortliffe, Feigenbaum, Buchanan, and Genesereth). It was Stanford's active ongoing research in medical computer science, plus a world-wide reputation for the excellence and rigor of those research efforts, that persuaded the University that the field warranted a new academic degree program in the area. A group of faculty from the medical school and the computer science department argued that research in medical computing has historically been constrained by a lack of talented individuals who have a solid footing in both the medical and computer science fields. The specialized curriculum offered by the new program is intended to overcome the limitations of previous training options. It focusses 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 a second class of five entered last summer. A third group of seven students has just been selected to begin during 1985. The proposed steady state size for the program (which should be reached in 1986) is 20-22 trainees. Applicants to the program in our first two years have come from a number of backgrounds (including seven MD's and five medical students). We do not wish to provide too narrow a definition of what kinds of prior training are pertinent because of the interdisciplinary nature of the field. The program has accordingly encouraged 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.

With the exception of this latter group, 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 in this application.

Early evidence suggests that the program already has an excellent reputation due to:

- high quality students, many of whom are beginning to publish their work in conference proceedings and refereed journals;
- 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 (almost $1M in hardware gifts from Hewlett Packard, Xerox, and Texas Instruments; over $200K in cash donations from corporations and foundations; and an NIH post-doctoral training grant from the National Library of Medicine).

The *Master of Science in Computer Science: Artificial Intelligence (MS:AI)* program is 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 and project involvement. During the second year, students implement and document a substantial AI application project.

2.1.4.8. Resource Community Management

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 is described in some detail in Section 2.3.3 on page 181 of the renewal plan. It has continued to function effectively as summarized below.

- The AIM Executive Committee meets regularly by teleconference to advise on new project applications, discuss resource management policies, plan workshop activities, and conduct other community business. The Advisory Group meets together at the annual AIM workshop to discuss general resource business and individual members are contacted much more frequently to review project applications. (See Appendix C on page 307 for a current listing of AIM committee membership).

- We have actively recruited new application projects and disseminated information about the resource. The number of formal projects in the SUMEX-AIM community still runs at the capacity of our computing resources. With the development of more decentralized computing resources within the AIM community outside of Stanford (see below), the center of mass of our community has naturally shifted toward the growing number of Stanford applications and core research projects. We still, however, actively support new applications in the national community where these are not able to gain access to suitable computing resources on their own.

- With the advice of the Executive Committee, we have awarded pilot project status to promising new application projects and investigators and where appropriate, offered guidance for the more effective formulation of research plans and for the establishment of research collaborations between biomedical and computer science investigators.

- We have allocated limited "collaborative linkage" funds as an aid to new projects or collaborators with existing projects to support terminals, communications costs, and other justified expenses to establish effective links to the SUMEX-AIM resource. Executive Committee advice is used to guide allocation of these funds.

- We have carefully reviewed on-going projects with our management committees to maintain a high scientific quality and relevance to our biomedical AI goals and to maximize the resources available for newly developing applications projects. Several fully authorized and pilot projects have been encouraged to develop their own computing resources separate from SUMEX or have been phased off of SUMEX as a result and more productive collaborative ties established for others.

- We have continued to provide active support for the AIM workshops. The last one was held at Ohio State University in the summer of 1984 and the next one will be in Washington, DC, hosted by the National Library of Medicine under Drs. Lindberg and Kingsland.

- We have continued our policy of no fee-for-service for projects using the SUMEX resource. This policy has effectively eliminated the serious administrative barriers that would have blocked our research goals of broader scientific collaborations and interchange on a national scale within the selected AIM community. In turn we have responded to the correspondingly greater responsibilities for careful selection of community projects of the highest scientific merit.

- We have tailored resource policies to aid users whenever possible within our research mandate and available facilities. Our approach to system scheduling, overload control, file space management, etc. all attempt to give users the greatest latitude possible to pursue their research goals consistent with fairly meeting our responsibilities in administering SUMEX as a national resource.

As indicated above, we have sought to retain SUMEX resources for new projects, those exploring new areas in biomedical AI applications and those in such an early state of feasibility that they are unable to afford their own computing resources. This policy has worked effectively as seen from the following lists of terminated projects and projects now using their own computing resources at other sites:

Projects Moved All or In Part to Other Machines:

Stanford Projects:

- GENET [Brutlag, Kedes, Friedland - IntelliCorp]

National Projects:

- Acquisition of Cognitive Procedures (ACT) [Anderson - CMU]
- Chemical Synthesis [Wipke - UC Santa Cruz]
- Simulation of Cognitive Processes [Lesgold - Pittsburgh]
- PUFF [Osborne, Feigenbaum, Fagan - Pacific Medical Center]
- CADUCEUS/INTERNIST [Pople, Myers - Pittsburgh]
- Rutgers [Amarel, Kulikowski, Weiss - Rutgers]
- MDX [Chandrasekaran - Ohio State]
- SOLVER [P. Johnson - University of Minnesota]

Completed Projects Summary

Stanford Projects:

- DENDRAL [Lederberg, Djerassi, Buchanan, Feigenbaum]
- MYCIN [Shortliffe, Buchanan]
- EMYCIN [Shortliffe, Buchanan]
- CRYSALIS [Feigenbaum, Engelmore]
- MOLGEN I [Feigenbaum, Brutlag, Kedes, Friedland]
- AI Handbook [Feigenbaum, Barr, Cohen]

- AGE Development [Feigenbaum, Nii]

National Projects:

- Ventilator Management [Osborne, Feigenbaum, Fagan - Pacific Medical Center]

- Higher Mental Functions [Colby - USC]

2.2. Planned Resource Activities

We have already summarized the overall aims of the SUMEX-AIM resource for the proposed 5-year renewal period on page 64. This section gives details of our research plans in pursuit of those aims for the major areas of our resource activities -- core research and development, collaborative research, service, training and education, and dissemination. To recap the overall scope and guiding goals of our new work:

- SUMEX-AIM is a national computing resource that develops and provides advanced computing facilities and expertise to support 1) a long-term program in basic research in artificial intelligence, 2) applying AI techniques to a broad range of biomedical problems by collaborative and user projects at Stanford and other universities around the country, 3) studying and developing methodologies for disseminating AI systems into the biomedical community, 4) experimenting with communication technologies to promote scientific interchange, and 5) developing better tools and facilities to carry on this research.

- Our applications, core research, and system development will be directed toward realizing and exploiting the computing environment that will be routinely available in the late 1980's and early 1990's, based on compact, decentralized, high-performance personal workstations that take advantage of the intelligent computing environments beginning to emerge from today's Lisp workstations. Consistent with these plans, we will immediately discontinue DRR subsidy for the DEC 2020 demonstration machine and for the shared VAX 11/780 time-sharing system. Also we will gradually and responsibly phase out DRR support for the DEC 2060 mainframe system that has been our chief shared resource and link to the past.

- There are consistent threads through our applications, system dissemination, core research, and computing environment development work. These threads are that our research work at all levels is driven by the real-world scientific applications that we undertake; that we choose applications that have a high impact on current medical and biological problems and that expose key underlying AI research issues; and that we seek to maximize the availability of the facilities for and results of this work in the biomedical community. This is seen, for example, in the coupling between our core research and development work and applications projects such as ONCOCIN and PROTEAN.

- We must continue to provide the computing resources for the growing Stanford biomedical AI research community and the national projects still dependent on us, to emphasize nurturing newly started AI applications, to serve as a communications cross-roads for the large and diverse AIM community, and to ensure broad dissemination of our research results and methods.

2.2.1. Core Research and Development

Reasoning in medicine and the biological sciences is knowledge-intensive. A recent article in *Science* [12], for example, discusses the role of information in the search for a cure for cancer. As the rate of explosion of knowledge continues to increase, clinicians and biomedical scientists must turn to computers for help in managing the information, and applying it to complex situations.

Artificial intelligence 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 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. 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. But the current ideas fall short in many ways, necessitating extensive further basic research efforts. Our core research goals are to analyze the limitations of current techniques, to investigate the nature of methods for overcoming them, and to develop tools to build and disseminate new and more effective biomedical expert systems.

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.

2.2.1.1. ONCOCIN-Related Core Research

As mentioned earlier in this application, our research plan for the next five years includes merging the core research activities of the ONCOCIN project with other basic research activities coordinated by the SUMEX resource. The ONCOCIN project is now in its sixth year and has involved approximately 40 research staff and students, some of whom have worked full time on aspects of the program or its knowledge base. It is accordingly large and has elements that span a variety of basic and applied research issues. The project's elements have been summarized in some detail elsewhere in this application and in the SUMEX annual report.

Since 1983 the Biomedical Research Technology Program, through a resource-related grant (RR-01631), has supported the effort to convert ONCOCIN to run on professional workstations (the Xerox 1108 Lisp machine). When that grant terminates in 1986, ongoing research will include a mixture of applied activities (evaluation of the workstations in the Stanford clinic and experiments to implement ONCOCIN workstations in private oncology offices in Northern California) and more basic activities intended to generalize past ONCOCIN results for the AIM community. We propose to continue the basic aspects of this work as core research under the SUMEX grant, and use complementary support for the other aspects of the project from the National Library of Medicine and, if a pending application for a dissemination experiment is successful, jointly from the National Center for Health Services Research and the National Cancer Institute.

In this section we summarize the core research activities that we intend to pursue in the context of ONCOCIN. They fall into four principal categories: implementation of ONCOCIN workstations in the Stanford clinic, knowledge acquisition research (OPAL), research to generalize ONCOCIN for application in clinical trial domains other than medical oncology (E-ONCOCIN), and research on generalized approaches to strategic therapy planning (ONYX).

Background on The ONCOCIN Program

From the outset, the ONCOCIN research effort has been directed towards both basic research in artificial intelligence and the development of a clinically useful consultation tool. We initially sought to apply techniques developed during our earlier work on the MYCIN system and to extend those methods to interact with a large database of clinical information. More recently, however, the system has departed from the uniform production rule approach of MYCIN in several significant ways (e.g., introduction of heterogeneous knowledge structures and distributed control processes [50] in the workstation version of ONCOCIN). Our approach to these problems has been greatly influenced by the Lisp machine technology to which we were first exposed through the foresight of SUMEX when it acquired such experimental machines in the early 1980's.

The initial version of ONCOCIN, including its clinical implementation in our cancer clinic, runs on a time-shared DEC-20 computer and uses a customized video display terminal installed in our oncology clinic. Since May of 1981, the prototype has been used on a limited experimental basis by oncology faculty and fellows to obtain advice on the treatment of patients enrolled in protocols for the treatment of Hodgkin's disease and non-Hodgkin's lymphoma. In the past year, additional protocols for adjuvant chemotherapy of breast cancer were added to the system.

We are excited by the promise of this prototype version of ONCOCIN. Formal evaluation of the system has shown that ONCOCIN does very well in suggesting therapy, even in cases where complex attenuation or changes in drugs are required [33]. It has also had a significant effect on the completeness with which clinical trial data are captured and made available for analysis [35]. In addition, we are extremely encouraged by the effectiveness of the interface program we have devised (the *Interviewer*) and the speed with which new users have been able to learn to use the system.

We believe that our current efforts to adapt the existing prototype for use on professional workstations will increase ONCOCIN's clinical acceptability. The use of a dedicated computer featuring high resolution graphics and mouse pointing devices to obviate typing should make the system even more attractive to busy physicians. As is described in the ONCOCIN progress report elsewhere in this proposal, we expect to have two Lisp machine (Xerox 1108) workstations in use in the Stanford oncology clinic by mid-1986. Thus, the continuation of ONCOCIN research in that clinic (knowledge base enhancement, software development in response to user feedback, and evaluations of the impact and acceptance of the workstation technology) will continue under the SUMEX umbrella after the merger of the SUMEX and ONCOCIN activities at the beginning of the next grant period. We should emphasize that, because of the moderate price of these computers, we look forward to transferring ONCOCIN for use in small clinics and physicians' offices. This will offer private physicians up-to-date decision support for the treatment of cancer patients (a recognized area of need) while allowing randomized clinical trials (RCTs) in oncology the benefit of greatly expanded access to appropriate patients. A four year experiment to install and test ONCOCIN in private offices has been proposed and is awaiting review and a site visit at this time.

Automated Knowledge Acquisition for RCTs

RCTs are based on rigidly structured therapy plans. Oncology protocols demonstrate this point nicely. RCT protocols are comprised of treatment *arms*, which in the case of oncology specify sequences of chemotherapy or radiotherapy. There is an explicit hierarchy of knowledge elements in these protocols which becomes important for knowledge acquisition. The hierarchy for a typical cancer chemotherapy protocol is shown in Fig. 6.

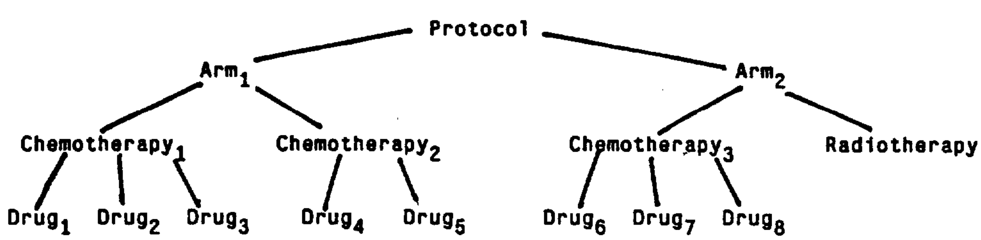

Figure 6: Sample Chemotherapy Protocol Hierarchy

ONCOCIN uses a variety of internal representations to store protocol knowledge. For example, in one arm of a protocol for small cell lung cancer, seven different drugs are used as part of two chemotherapies in a specific sequence over seven weeks. The sequence of chemotherapies is repeated five times, making the total duration of treatment 35 weeks. The names of the chemotherapies are POCC and VAM. Administering POCC requires that the patient make two separate clinic visits to receive medication during each treatment cycle. Hence, POCC is divided into two sub-cycles: POCC-A and POCC-B. After the second complete cycle of POCC, the patient is given cranial irradiation. The computer representation of this entire complex sequence is:

```
((POCC 1 A) (POCC 1 B) (VAM 1)
 (POCC 2 A) (POCC 2 B)
 (XRT CRANIAL)
 (VAM 2)
 (POCC 3 A) (POCC 3 B) (VAM 3)
 (POCC 4 A) (POCC 4 B) (VAM 4)
 (POCC 5 A) (POCC 5 B) (VAM 5))
```

This purely procedural knowledge can be extracted from protocol documents fairly easily; one need not understand oncology. However, much of the important knowledge in ONCOCIN is more judgmental and is represented in the form of production rules. ONCOCIN currently uses over 400 rules to determine:

- how to adjust specific drug dosages because of treatment-induced low blood counts or other adverse (toxic) reactions to therapy
- when to delay treatment or abort a therapy cycle
- how to modify therapy in light of a patient's changing clinical conditions or response to the protocol
- when to order certain laboratory tests and how to interpret their results.

Note that these issues are generic for all clinical trials, and similar rules could be written to assist with proper administration of treatment for RCTs in other medical domains.

An example of one such rule, drawn from the ONCOCIN system, is shown in Fig. 7. It was developed by examining a formal protocol and then further enhancing and validating the knowledge through discussions between an oncologist and a knowledge engineer.

```
To determine the current attenuated dose for patients with all lymphomas
      in CHOP chemotherapy for Cytoxan or Adriamycin:

If:  1. The blood counts warrant dose attenuation
     2. It patient did not receive chemotherapy
          before the last radiation therapy
     3. This is the first cycle after significant radiation
     4. This is not the first visit after an Abort cycle

Then:  Conclude that the current dose is 75% of the standard
       dose further attenuated by either the dose attenuation
       for low WBC or the dose attenuation for low platelets,
       whichever is less.
```

Figure 7: Sample ONCOCIN Rule, Translated to English from Internal Format

The knowledge engineer then must convert this rule into a representation understandable by the computer. The rule format for computer use is generally unreadable to the clinician who helped to develop the rule in the first place. It is the translation shown in the figure that is created and reviewed by the clinician. The knowledge engineer's detailed understanding of the manner in which information is represented in the computer allows him or her to develop the corresponding machine-understandable format.

Because the knowledge engineering process is cumbersome and inefficient, we have recently embarked on work to develop a system, termed OPAL, that acquires new knowledge of oncology protocols directly from physicians while shielding them from technical details. As part of our SUMEX core research activities, we will seek to generalize this approach for application in other medical domains in which RCTs are commonly used. The knowledge contained in protocols for oncology (and for other RCTs as well) has already been formalized in the protocol document. The most fundamental problems of conceptualizing and structuring the domain knowledge should therefore not be an issue in this work.

For example, detailed discussions with our oncology experts and review of dozens of protocol documents make it clear that the knowledge in protocols is both predictable and constrained by the very nature of oncologic clinical trials. For each concept that appears in oncology protocols, we can anticipate the general nature of most of its possible *values*. For example, we can assume that all drugs will have a *dose* that can be represented by an integer. All drugs will have a *route*--intravenous, intramuscular, or oral. Our knowledge of the field allows us to determine *a priori* what possible choices might be appropriate for most concepts. This has great implications for automated assistance in knowledge acquisition.

We have known for some time that it would be ideal to provide an environment so that the physicians can themselves enter and manipulate knowledge of a RCT protocol and related medical knowledge. However, since it is generally unrealistic to teach collaborators to become programmers or knowledge engineers, we are faced with the traditional problems of getting a computer to understand the *meaning* underlying unstructured phrases or sentences entered by a physician. TEIRESIAS had approached the problem by cleverly manipulating the context of an interaction with an expert, thereby simplifying the task of understanding entries [13]. However, problems in computer-based understanding of natural language (still a major research topic in artificial intelligence) prevented TEIRESIAS from becoming sufficiently robust for routine use. We have been unwilling to reopen the Pandora's box of natural language understanding for the ONCOCIN project, and therefore in the early years have had to resort to the LISP-based entry of knowledge.

Two factors have accounted for our decision to turn again to the problem of knowledge acquisition. The first has been a simple matter of *need*. As we have developed plans to adapt ONCOCIN for use on single-user machines in physicians' offices, and have contemplated the large numbers of protocols that must be available online for practical use of such a tool, we have been forced to acknowledge the necessity of an enhanced knowledge acquisition capability. Second, in transferring ONCOCIN to personal workstations and familiarizing ourselves with this new technology, we have become aware of the potential for using advanced graphics techniques to avoid problems of natural language understanding during entry of knowledge by a computer-naive user. To explore the possible use of the graphics capabilities of LISP machines to facilitate knowledge acquisition directly from experts, we have recently developed a prototype system for knowledge entry. OPAL was designed in close collaboration with oncologists who will be the eventual end users of such a system. To build the prototype version of OPAL we reviewed all of the concepts that had been required for each of the protocols that we entered by hand, and explored a large number of existing protocol documents that we hoped to enter into the completed system.

The OPAL prototype runs on the same professional workstation (the Xerox 1108 "Dandelion") on which the new version of ONCOCIN is being developed. Like the new ONCOCIN system, OPAL is designed to take advantage of the advanced graphics capabilities of the workstation and uses a mouse pointing device almost exclusively for input by the physician.

In developing OPAL, we attempted to organize the information to be entered by the physician in a manner similar to the structure of typical protocol documents. A constant consideration was to request knowledge from the physician in a manner consistent with the way oncologists tend to think about protocols. OPAL guides protocol entry in a loose fashion; the expert is provided with an ability to change topics at his or her convenience. However, the program follows an orderly progression, first asking for general information about the scope of the protocol, principal investigators, and inclusion and exclusion criteria; next asking for the protocol "schema" -- a shorthand notation that describes the sequences of treatments; and finally requesting information on specific drugs, dose modifications, and diagnostic tests required by the protocol.

The questions for each of these categories are grouped into individual windows on the graphics display. These windows contain a number of "blanks" on the screen to be completed in order to provide pertinent protocol information. Most blanks can be filled in by selecting them with the mouse and then selecting an item from a menu that is displayed. Rarely the blanks are filled in by typing at the keyboard. The windows are not all displayed at once but rather are selected one at a time by the physician working his or her way through a protocol. Selecting a window brings it "into view". In the present OPAL prototype, most of the major windows are portrayed graphically as a stack of overlapping "file folders" on the screen. Using the "mouse" to select the "tab" of one of these folders brings the corresponding window into view. Special menu windows can be created for the entry of purely numerical data. For example, we have developed menus, called "registers", that appear either in the format of a 10-key calculator pad (for free-form digit selection) or else in a columnar format, akin to the front of an old-style cash register. In either case, the user indicates the appropriate digits sequentially using the mouse without needing to touch the keyboard. Several examples of the windows used for protocol entry are provided in the working paper by Differding included as an Appendix to this application.

The OPAL prototype presumes that the user will have no appreciation for how knowledge is stored in the computer for use by the reasoning elements in ONCOCIN; the user need only be able to understand oncology protocol documents. The system deals with chemotherapy knowledge at such a high level that the user is completely shielded from issues of knowledge base organization and format. The physician using OPAL needs to be concerned only with the actual knowledge in the protocol to be entered.

The preliminary version of OPAL consists of a series of windows that may be displayed on the screen of the 1108 workstation in any order. Each window represents a series of questions or blanks to be filled in for a specific portion of a protocol's knowledge. For example, one window asks questions about the names and standard dosages for the drugs to be used for a given chemotherapy; another asks what laboratory studies are required by the protocol; a third inquires what actions to take if certain toxicities develop.

For each possible "blank" in the window, information is entered automatically by the system if the corresponding data are already known because of previous responses (e.g., if a standard chemotherapy is chosen in one window, the individual drugs involved will then appear in all of the other windows that ask for drug information). Otherwise, selecting a blank with the mouse causes a menu with possible completions for that item to "pop up" on the screen. The mouse is then used to select the desired response from the menu.

The OPAL prototype has been tested by several physicians and all have found the system easy to use after a few minutes of training. Frequent feedback from our oncology collaborators has allowed us to make modifications, expanding the options in certain menus and improving the user interface. These modifications have been effected by reprogramming parts of the system. However, we plan to be able to make changes to OPAL eventually by editing *data structures*, rather than by having to update the actual computer *programs*.

When a protocol is entered using OPAL, the knowledge ultimately must be encoded in an internal form so that ONCOCIN can use it to give advice and manage the protocol data. We see this encoding occurring in a two stage process, with an intermediate data structure serving to insulate the interaction with OPAL from the detailed structure of the knowledge base. Thus OPAL will be used to enter protocol knowledge, it will be stored in an intermediate data structure (or IDS), and then further refined into a knowledge base for use by ONCOCIN. As is outlined in the next section, these ideas generalize to RCT advice systems in other clinical domains -- a generalized OPAL might be used to enter RCT guidelines, thereby creating a knowledge base for use by a generalized version of ONCOCIN.

Generalization of ONCOCIN: *E-ONCOCIN*

Most protocols in clinical medicine contain elements in common with oncology trials. We plan to build on our experience creating OPAL to apply the same methodology to develop expert systems for RCTs in other medical areas. This research to develop generalized knowledge acquisition programs like OPAL for other RCTs will be of great practical importance. However, we recognize that the work will address significant theoretical issues in the field of medical artificial intelligence. In fact, we expect that the Meta-OPAL work outlined below will constitute a Ph.D. dissertation for one of our Medical Information Sciences graduate students (Dr. Mark Musen).

What we propose is a high-level tool for use by knowledge engineers in conjunction with clinicians to define all the properties of a *knowledge acquisition system* (KAS) that may be used subsequently to enter the knowledge for a particular *class* of clinical trials. OPAL is an example of a KAS, one that is customized for the class of clinical trials relevant to clinical oncology. A KAS for another domain, such as hypertension or epilepsy management, might look very different. Certainly the display windows for protocol entry would bear little resemblance to those used in the current version of OPAL. This new high-level tool, *Meta*-OPAL, will take as its input the complete specifications for a KAS. It will produce as its output a data structure that will enable a second program, E-OPAL, to interact with a domain expert to capture and encode a whole class of new protocols. These encoded protocols can then be used for data management and consultation by a domain-independent version of ONCOCIN (the ONCOCIN inference engine, to be termed E-ONCOCIN)[1]. E-OPAL will be a version of OPAL stripped of all its built-in oncology knowledge. E-OPAL thus will rely on Meta-OPAL to provide all the information required to perform knowledge acquisition and management. The relationships of the various modules is diagramed in Figure 8.

The concept of a "knowledge acquisition system for knowledge acquisition systems" is attractive in many respects. First, many of the problems of a limited "world view" in a program such as OPAL will be readily overcome because all of the domain assumptions (e.g., beliefs about oncology, cancer protocols, or chemotherapy) will be explicitly declared at the Meta-OPAL level. For example, an implicit assumption built into the present OPAL prototype is that patients are treated with either chemotherapy or radiotherapy. The physician using OPAL is never asked to enter information regarding, say, *surgery* because knowledge about options for surgery is not currently within OPAL's "world view". Even by modifying OPAL to specify new parameters, no protocol that called for repeated surgical procedures could be satisfactorily encoded unless we had an ability to make even higher-level modifications to OPAL.

At present, we can make this sort of higher level modification to OPAL only by

[1]The names E-OPAL and E-ONCOCIN are inspired by the similar domain independent tool developed by our group in the 1970's. This program, EMYCIN or "Essential MYCIN", is the inference engine separated from the knowledge base of MYCIN

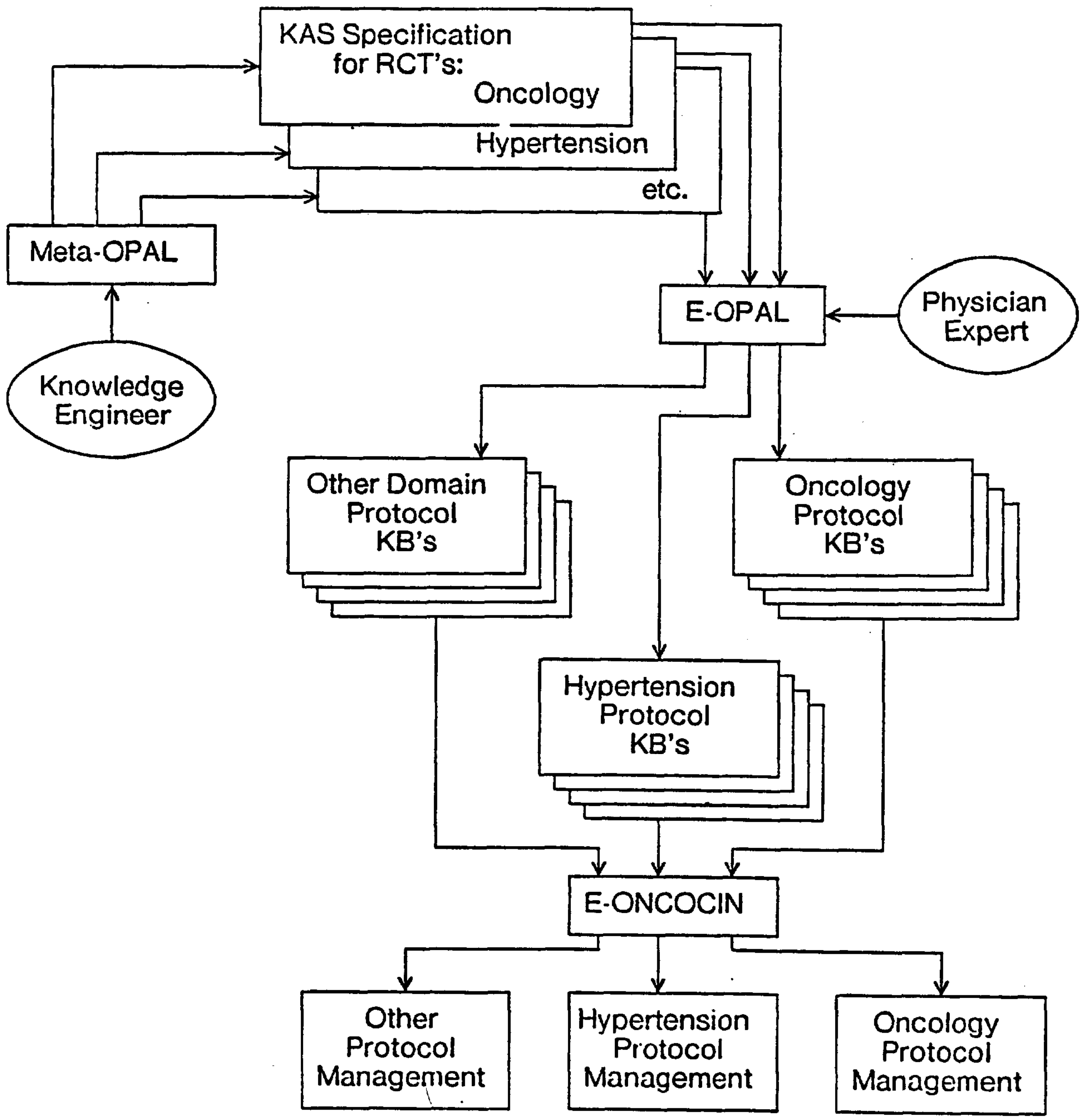

Figure 8: Components of the Meta-OPAL System

reprogramming the whole system and revising the format of the Intermediate Data Structure (IDS). However, if the operation of OPAL itself were completely and explicitly defined in a very high level syntax, it would be far preferable to use Meta-OPAL to revise this representation, automatically creating a new knowledge acquisition system (KAS) that would incorporate (1) the necessary windows and menus, (2) an appropriate IDS format, and (3) specifications for how the IDS should be translated into E-ONCOCIN knowledge bases.

Meta-OPAL and the necessary KAS definition language will allow us maximum flexibility in adopting OPAL for unusual protocols that might be encountered in the future. If the KAS definition language is general enough, it will allow knowledge acquisition for clinical trials outside of the domain of oncology. Because the ONCOCIN inference engine (E-ONCOCIN) makes no specific assumptions about oncology, Meta-OPAL could produce knowledge acquisition systems that would permit physicians to enter new protocols for any kind of clinical trial; E-ONCOCIN could then be used for patient consultations and for data management.

The Domain of Clinical Trials is Well Suited for Meta-OPAL:

Just as the structure of knowledge within clinical protocols is generally easy to anticipate, knowledge *about* clinical protocols is equally predictable. For example, the sequence of interventions to take in any clinical trial should always be representable as a schema. This schema might be similar in syntax to that now used by OPAL to express the order of treatments in cancer protocols. Other representations might be more appropriate for RCT's in different domains. In oncology,

```
CHOP x 6
```

is quite satisfactory. However, stepped care for hypertension might be better expressed using a format such as:

```
STEP 1: Hydrochlorothiazide
STEP 2: add Labetolol
STEP 3: add Captopril
```

Our initial work on Meta-OPAL will include developing a complete and unambiguous syntax for specifying protocol schemas. Part of the KAS definition language will involve declaring formatting options and what entries are permissible when a schema is entered into the resultant KAS.

Knowledge about clinical trials is predictable in other ways. For instance, all protocols list a host of laboratory test results and clinical conditions that must be recorded and that may cause an alteration in the treatment plan. The number of ways in which therapy may be modified within a given class of protocols is finite; these kinds of actions will have to be specified in the KAS definition language.

Knowledge acquisition systems for RCTs also can capitalize on another constraint in their domain: patients with concurrent diseases that might complicate analysis of the study are excluded from participating in protocols. The scope of the knowledge needed for a given expert system can therefore be limited to the one disease under investigation. The task of designing a KAS for a given class of clinical trials is clearly simplified when the scope can be focused in this way.

Although there are many different kinds of clinical trials, knowledge about such studies is always formalized in a protocol document. Examining protocol documents will allow us to generalize about what characteristics are required for knowledge acquisition systems in each of the domains studied and provide the basis for developing Meta-OPAL and its KAS definition language. Our experience in developing and using OPAL will also be essential in guiding our design for Meta-OPAL.

How Meta-OPAL Will Work

The user interface to Meta-OPAL will involve the same menu-driven approach we have adopted in the design of OPAL. Many of the usual difficulties of communicating with the computer will again be obviated by the use of graphics-based editing.

The knowledge engineer, in conjunction with a physician-expert, will use Meta-OPAL to specify the general nature of the clinical trials involved (e.g., sequential therapy, stepped care) and certain study design issues (e.g., cross-over trials, repeated randomizations). The expected modalities of treatment will also be declared. Meta-OPAL can then establish the schema syntax for the protocols to be entered.

The program will then assist the user in structuring the IDS. The names and meaning of each of the IDS entries will then be declared. The relationships among the various IDS components will be specified using graphics.

A list of all knowledge base *parameters* will then be declared and the same rule definition language that we will develop for OPAL will be used to specify how the rules to conclude each of the parameters can be generated from the IDS. Parameters will fall into several categories (e.g., clinical conditions, concluded drug dosages, intermediary parameters used in the reasoning process) and the nature and use of each parameter will have to be specified. For example, the user must specify which parameters correspond to items that must appear on a patient's "flow sheet" when displayed by E-ONCOCIN. Similarly, it will be necessary to indicate those parameters whose values will represent the system's "recommendations" during a consultation.

Finally, Meta-OPAL will prompt the knowledge engineer with the basic information needed for each of the windows that will appear in the completed KAS. The exact window formatting will then be entered by selecting locations on the screen with the mouse and typing in the text that should appear there when the KAS is generated. If dissatisfied with the location of particular blanks, the knowledge engineer will be able to use the mouse to rearrange the formatting. For each blank, the system will ask the knowledge engineer to specify the corresponding menu that will appear when the blank is selected by the KAS user. The knowledge engineer must also indicate where the entry for the blank is to be stored in the IDS and any information needed to check for completeness or consistency.

Once the user has completed entry of information into Meta-OPAL, a new data file will be created that will contain all of the specifications of the KAS. This file will serve as input to the E-OPAL program, which will follow the file's guidance in displaying windows and gathering data during the knowledge acquisition process. The information will be in a format that can be modified by a standard text editor, if necessary, as well as by Meta-OPAL. The knowledge will be encoded using a KAS definition language.

KAS Definition Language:

We will limit the scope of representations expressible in the KAS definition language to the area of knowledge acquisition for clinical trials. This not only makes implementation of Meta-OPAL more feasible, but restricting the scope of the system will also make the finished program easier to use because the necessary input will be more focused. The kinds of knowledge contained in this output from Meta-OPAL should be apparent from our previous discussion of how Meta-OPAL will work. The syntax we will develop must express a number of different concepts:

1. Various definitional items must be specified to the system. For each kind of knowledge acquisition system Meta-OPAL can create, we must have a syntax for declaring the names and the properties of:

a. The *modalities* of treatment; for example, oncology protocols involve chemotherapies and radiation. A protocol for treating esophageal varices might use various surgical or endoscopic procedures as modalities.

b. The *agents* of treatment; for example, the three drugs vincristine, adriamycin, and methotrexate are the agents used in modality VAM in cancer chemotherapy. "Positive reinforcement" and "negative reinforcement" are two agents of the modality "behavior modification" that could be used in psychiatry protocols.

c. *Standard toxicity grades and their text definitions*, representing various measures of adverse effects on organ systems. Each toxicity grade would also be linked to a *parameter* so that E-ONCOCIN would be able to draw conclusions based on the presence or absence of certain adverse conditions.

2. The list of parameters and their associated properties must be indicated, including rule definition language specifications on how to generate the rules that may conclude each parameter's value. The types of parameters include:

   a. Physical examination findings

   b. Laboratory tests and test results

   c. Clinical conditions, such as "no evidence of disease", "complete response", or "progressive disease"

   d. All "conclusions" reached by the system, including final treatment recommendations.

3. We must permit specification of all of the various *actions* one might take to change any component of the treatment plan. Such actions could involve alteration of the protocol at any level. For example, the protocol itself could be terminated or extended. Administration of any of the modalities of treatment might be delayed or canceled. The dosages of any of the therapeutic agents might be changed, or new agents might be substituted.

   Other actions that do not specifically modify therapy need to be declared. For example, based on some set of parameters, one might want to "order a lab test" or "notify the principle investigator" of some problem.

   Each of these actions will appear as potential entries in portions of the IDS and will accordingly be specified in menus in the resulting KAS (i.e., in menus displayed by E-OPAL as it takes its directions from a KAS file that was produced by Meta-OPAL). Such menus will offer steps to take in response to various values of defined parameters.

In addition to the domain knowledge, the KAS definition language will require declaration of important systems information, including:

1. A description of the high-level appearance of the knowledge acquisition system, including the contents and layout for each window and the nature of each blank and its corresponding menu. Meta-OPAL will determine this knowledge from the graphical inputs of the user when defining the KAS.

2. Specification of the necessary IDS to use for the specific E-OPAL

application, including the complete IDS format, the mapping of blanks from the various windows into the IDS, and the control information needed to translate the IDS into the final knowledge base for use by E-ONCOCIN.

E-OPAL:

In order for Meta-OPAL to produce new knowledge acquisition systems, we will first have to develop E-OPAL, a program that will capture the behavior of the present OPAL prototype. However, E-OPAL will acquire all of its formatting specifications for windows and menus from the output data file produced by Meta-OPAL, rather than from structures internal to the program itself. It will use the knowledge encoded in the KAS definition language to produce an IDS, transfer knowledge from display windows to and from that IDS, and use the IDS to produce a knowledge base for the ONCOCIN inference engine. The physician will enter protocol knowledge in E-OPAL in a manner identical to the present OPAL system.

E-ONCOCIN:

The current ONCOCIN system has been written with care to keep the ONCOCIN knowledge base separate from its inference engine. Thus a relatively complete version of E-ONCOCIN already exists, and this separation is being further refined as part of our translation of ONCOCIN to run on the 1108 workstation. However, we anticipate further changes as our understanding of the IDS and Meta-OPAL evolve.

Encoding New Protocols with Meta-OPAL:

We will test the Meta-OPAL system by rewriting OPAL using Meta-OPAL. This will be accomplished by producing a knowledge acquisition description file using Meta-OPAL and showing that E-OPAL, driven by Meta-OPAL's output, produces a knowledge acquisition system with behavior grossly identical to that of OPAL. This will produce a more generalizable version of OPAL that overcomes some of the limitations of the initial prototype.

We will also use the system to encode protocols in at least one (and possibly two) other medical domain. Dr. Peter Rudd, a member of the Division of General Internal Medicine at Stanford, is conducting randomized controlled trials of new antihypertensive medications and has agreed to collaborate on knowledge base development. This domain of hypertension and its treatment will provide a useful environment for testing the definition of new knowledge acquisition systems using Meta-OPAL. In addition, Dr. Gordon Banks from the University of Pittsburgh (a member of the INTERNIST/CADUCEUS project) has approached us about adapting ONCOCIN for us in protocol-directed management of epilepsy patients. This may well provide another pertinent domain for testing the generality of the notions described here.

Strategic Therapy Planning

ONYX is an ONCOCIN-related subproject designed to fill the need for planning in application areas where traditional planning methodology is difficult to apply. While the program is being developed to assist with the planning of cancer therapy, its architecture is intended to be of use whenever goals are ill-specified, plan operators have uncertain effects, or trade-offs and unresolvable conflicts occur between parts of the goal. ONYX combines strategic "rules of thumb" with a mechanistic model of the domain to determine a set of plausible therapy plans. This is accomplished with a three step process: (1) generate a small set of plausible plans based on current data; (2)

simulate those plans to predict their possible consequences; and (3) based on the results of those simulations, rank the plans according to how well each meets the goals for the situation.

Much of the early work in artificial intelligence techniques for planning made simplifying assumptions about the various choices that can be made at each step of the plan, and in representing the effects of each of these planning steps. In medicine, the planning task often cannot be represented in a form useful to a conventional planning program. Often the goals are ill-specified and the operators have uncertain effects. Furthermore, incomplete and unresolvable interactions occur between the parts of the goal, limiting the usefulness of some of the techniques developed least commitment and plan repair techniques. Consequently, medical therapy planning programs such as VM [17], ONCOCIN, and ATTENDING [49] have frequently relied on algorithms or step-by-step protocols to provide explicit guidelines in the construction of plans appropriate to a particular patient's condition.

Our work with ONCOCIN has revealed an important limitation of medical planning systems which use explicit criteria such as algorithms and protocols. The knowledge in these specifications is a "compiled" version of pathophysiological knowledge of the human body, and of the strategic knowledge of the domain. In ONCOCIN, plan elements are selected strictly according to the characteristics of the current treatment situation without considering the causal mechanisms of the domain or many of the strategies useful in prescribing therapy. Consequently, when a situation arises for which the algorithmic knowledge does not apply, the planning system often recognizes the problem, but cannot plan alternative therapy. The ONYX system is designed to suggest expert quality therapy plans in such difficult cases.

The planning process used by ONYX consists of three steps:

1. *Plan generation.* Using current and past data about the patient, and exploiting the hierarchical nature of possible plan steps, generate a small set of "plausible plans" which are consistent with the patient's current state and the treatment goals for the patient.

2. *Qualitative simulation.* Using causal knowledge of the human physiology, and of this patient's in particular, predict the future states of the patient if each of the plausible plans were in fact executed.

3. *Plan Ranking.* Using knowledge about how patient data satisfy the goals for the patient's progress, rank each of the plausible plans according to the extent that the simulation's predictions for each plan meet the therapy goals.

Cancer treatment strategies are often general statements such as "Try to give a greater quantity of therapy during the early stages of treatment". Restated in a particular context, this might indicate a preference for decreasing a drug dose to 75% rather than just 50% in response to a particular problem. Other strategies may be applied to a wide range of decisions in the plan generation process, from broad therapeutic choices (e.g. whether to give drug therapy or radiation therapy) to specific decisions about individual drug doses. One such strategy is: "If a problem is encountered with a treatment, try to eliminate the part of the treatment that might be causing the problem." In one context, this is interpreted as a suggestion to decrease or eliminate the previously administered drug that is the likely cause of toxicity. In another context, it may also be used to help decide between continued drug therapy and alternative treatments. Currently, such a strategy must be represented in each context in which it applies, rather than as a single more general principle.

The input to the planning process is the database of patient measurements (e.g., the size of the tumor, white blood count) collected over a number of prior treatment cycles. These input data are processed by the list of treatment strategies. The output of the plan generation phase is a set of possible treatment plans for the current patient visit.

The possible treatment plans are sent to the simulation component to determine the likely ramifications of the treatment. We have designed special software to allow for graphical description of the simulation model. The structure of the domain models is organized hierarchically according to *part-of* relationships. The behavior of a model is determined by the behavior and interconnections of its parts and by three knowledge bases which describe its behavior in response to stimuli. The state of a model is represented by a group of *state variables*, and by the states of its parts. Each model has *ports* through which it communicates with other models using message passing techniques provided by the object oriented system. Such hierarchical models can be built interactively on a Xerox 1108 LISP workstation.

The behavior of each model is described by three rule bases containing production rules. The first rule base dictates how a model will change its state according to the stimuli it receives through its ports from other models. The second rule base contains knowledge about how to make further conclusions about its state based on any recent changes. The third rule base dictates how the new state of the model will be propagated to neighboring models using a simple message passing scheme which acts along connections between models.

Simulation can provide information which the plan evaluation process can use to determine the likelihood that a plan will satisfy the goals for the patient. While the plan ranking phase of ONYX is still under development, early experiments indicate that the rule form used in the plan generation phase will provide some power in the ranking of plans after simulation. In addition, decision analytic techniques can be used to evaluate the decision trees developed by the strategic planning and simulation components.

E-ONYX:

We have thus far challenged and tested this developing system with only a single cancer protocol. However, we believe that the techniques can be expanded to other cancer protocols, and then to other types of clinical trials. We propose to generalize this program, with much of the work involved in representing the various types of plans that may occur among different clinical trial experiments. We expect that the form of the strategies may have to be modified for other medical trials. In addition, we need to verify that the hierarchical nature of the simulation process is sufficient to represent the dynamic processes as the treatment regimen of the clinical trial affects the body of the patient as well as the disease process.

2.2.1.2. Basic Research in AI

Overall Goals and Plans

Our basic AI research projects focus on understanding the roles of knowledge in symbolic problem solving systems -- its representation in software and hardware, its use for inference, and its acquisition. We are continuing to develop new tools for system builders and to improve old ones. The research crosses a number of application domains, as reflected in the subprojects discussed earlier, but the main issues that we are addressing in this research are those fundamental to all aspects of AI. We believe this core research is broadening and deepening the groundwork for the design and construction of even more capable and effective computer programs to aid in reasoning about biomedical problems.

As mentioned above, although our style of research is largely empirical, the questions we are addressing are fundamental. The three major research issues in AI have, since its beginning, been knowledge representation, control of inference (search), and learning. Within these topics, we will be asking the following kinds of questions. As our work progresses, we hope to leave behind several prototype systems that can be developed by others in the medical community.

In particular, we will focus on four areas with immediate coupling to biomedical applications problems and on several others that may have future application:

1. Blackboard Model of Reasoning -- can we design and construct a domain-independent framework for problem solving programs using the blackboard model and can we reason explicitly about control in that framework?

2. Constraint Satisfaction -- given a number of symbolic and numeric constraints defining a satisfactory solution to a problem, how can a problem solver efficiently find a solution?

3. Knowledge Acquisition -- how can knowledge-based programs effectively acquire the large amounts of domain-specific knowledge needed for high performance problem solving?

4. Qualitative Simulation -- how can biological modelling systems be constructed that use domain-specific knowledge to reason approximately about outcomes?

5. Other Research Areas -- architectures appropriate for highly concurrent symbolic computation, a retrospective on the AGE blackboard tool, logic-based systems, self-aware systems, and the SOAR general problem-solving architecture.

These major research themes are discussed in the subsections below and build upon the workstation and advanced computing environment technology also being developed under SUMEX core research.

1. Blackboard Model

GOALS

The long term goal of this part of our research is to improve the usability, the flexibility, and the inferential power of AI software systems for handling problems of hypothesis formation, signal understanding, constraint satisfaction and planning. We proposed to design and implement domain-independent tools for building complex reasoning systems within the blackboard framework. These include development aids as well as run-time utilities. In other research, we have a coordinated goal of applying the Blackboard framework as an organizing framework for parallel processing.

For the research described below, we have two main objectives. These are:

a. to develop scientific understanding of the "support environment" for Blackboard framework systems and of key tradeoff issues; to design tools for building systems; to implement a domain-independent system incorporating those tools.

b. to implement a substantial reasoning system in the BB1 framework in order to experiment with tradeoffs in the design. Specifically, we will work with the PROTEAN collaborative project to implement and experiment with the program that infers tertiary structure or proteins from NMR data (plus knowledge of primary and secondary structure). This work is described in the research plan for the PROTEAN project.

MOTIVATION

In building knowledge-based systems, we have come to understand the importance of flexibility in its operation. In the KSL, we have experimented with many frameworks for building systems including rule-based, frame-based, and logic-based frameworks. We have also experimented with various methods of inference and control, including goal-directed, data-directed, and opportunistic reasoning. Of the paradigms we know about, the one that seems to offer the most flexibility (at development time and run time) is the blackboard model of reasoning. It has not been as well studied or used as the rule-based or logic-based paradigms have been. Thus we believe a substantial research effort is warranted in order to understand its strengths and limits, and to build a suite of tools that allows us to experiment with it.

BACKGROUND

Though the Blackboard framework for problem-solving and hypothesis formation was conceived at Carnegie-Mellon during the DARPA Speech Understanding project in the early 1970's, it has received much of its scientific and practical development by scientists of our laboratory. The first post-CMU/HEARSAY development was in connection with the HASP system for passive sonar signal understanding. Subsequent efforts involved experiments with scientific applications (to x-ray crystallography), intelligence problems (ELINT and COMINT), and planning; as well as the development of the first software tool to assist knowledge engineers in constructing systems using the Blackboard framework (AGE-1).

As the last decade unfolded, the Blackboard framework was seen to be the most flexible and powerful set of software concepts we had encountered for organizing the processing of knowledge-based systems. It allowed arbitrary mixing of data-driven inference steps ("bottom up") with model-driven steps ("top down"). It allowed a hierarchy of levels of abstraction in the ongoing solution formation, from the most abstract (the global situation) to the least abstract (the supporting data or problem conditions). And it allowed multiple sources of knowledge to provide the links between these levels (i.e. supported information fusion).

The growing significance of the Blackboard framework has given importance to entering a second phase of its development: extensions of the basic concepts (e.g. reasoning from uncertain evidence) and extensions of the suite of software tools for building such systems.

BB1 [27] is a domain-independent environment for building AI systems in a "blackboard control architecture" [28]. Like the standard blackboard architecture [16], BB1 solves problems through the actions of independent knowledge sources that record, modify, and link individual solution elements in a structured database (the blackboard) under the control of a scheduler. It expands upon the standard architecture as follows:

1. It provides an interpretable, modifiable representation for knowledge sources with these attributes: event-based predicates for triggering; pattern-matching functions for identifying multiple triggering contexts; state-based predicates for assessing transient pre-conditions, and rule-based actions that instantiate prototypical blackboard modification templates. BB1 provides support facilities for knowledge source creation, modification, and checking.

2. Its blackboard representation permits dynamic assignment of attributes and values to objects on the blackboard and provides selective, demand-driven inheritance of attributes from linked objects, with local caching of results.

3. It provides explicit reasoning about control--the selection and sequencing of knowledge source actions--with control knowledge sources that construct dynamic control plans out of modular heuristics on a control blackboard. BB1 defines specific levels of abstraction and solution intervals for the control blackboard. It provides a vocabulary and syntax for expressing control heuristics. A simple scheduler decides which domain and control knowledge sources to execute by adapting to whatever control heuristics currently are recorded on the control blackboard.

4. It provides strategic explanation of problem-solving activities.

5. It provides generic learning knowledge sources to acquire new control heuristics automatically.

6. Its run-time user interface provides capabilities for: displaying knowledge sources, pending actions, and objects on the blackboard; graphically displaying partial solutions via a user-specified interface; recommending pending actions for execution; permitting a user to override a recommendation; executing a designated action; operating autonomously until a user-specified criterion is met.

BB1 is an evolving system incorporating the best results of several research activities. It currently is being used as a framework for the PROTEAN system here at Stanford and for several applications by other research and industrial organizations. We propose to continue developing BB1 as a prototype "next-generation" blackboard architecture.

RESEARCH PLAN

Trade-Off Between Knowledge and Control

As the complexity of the applications we attack increases, the tendencies have been to build more complex control structures. This is a natural consequence of a strategy of "divide-and-conquer" -- having broken the problem into manageable subproblems, the question arises as to how and when to bring the sub-problems together. The other factor that contributes to different control schemes is the difference in quality of knowledge that can be brought to bear at different points in the problem solving process. For example, if there is not much situation-specific knowledge to be applied at a particular point, a system can resort to a method of generating all possible solutions and testing them for credibility.

In the study of concurrent problem solving frameworks, control represents a serialization of knowledge applications. A preliminary study indicates that there can be a trade-off between knowledge and control. An almost control-free blackboard system may or may not converge to problem solutions. To date there is no research that addresses the trade-off possibilities between degrees of control and various kinds and amounts of knowledge. A blackboard architecture provides a very fertile medium in which this research can be conducted, because all information, including control information, is available on the blackboard. This provides an opportunity to vary the amount of utilization of the control information and control knowledge sources at the same time as adding and modifying task-specific knowledge.

Debugging at the Blackboard System Level

We propose to investigate what would constitute an effective suite of debugging aids for blackboard tools. This investigation will be based primarily on our experience in both using and building various blackboard tools.

The blackboard debugging aids that we will investigate include:

1. A blackboard break package. This package would permit, for example, execution-time insertion of conditional break-points for a specific type of modification of the blackboard nodes of a given class or classes, specific knowledge source invocation, and specific rule evaluation or invocation.

2. A blackboard inspector package. This inspector would permit the inspection of blackboard nodes and the relations between them at various levels of abstraction. These levels of abstraction might range from the entire blackboard presented as a graphics display of nodes by class icons with node relations represented by colored links to the detailed attributes and their values for a specific node presented as formatted text.

3. A stepper which would allow the single-step execution of a blackboard program at various levels of resolution, for example, event posting, knowledge source invocation and rule evaluation. This stepper could be turned off or on by the user or by the execution-time insertion of conditional stepper switch points.

4. A static analyzer which would analyze and present the relationships between, for example, event postings, knowledge source preconditions, knowledge source invocations, and possible blackboard node modifications.

We will use the results of this investigation to design and implement a suite of prototype blackboard debugging aids. Although these aids will be implemented in the context of a particular blackboard tool, for example, BB-1 or an AGE derivative, the underlying concepts should be applicable to a variety of blackboard tools. In particular, we plan to investigate how these debugging concepts could be extended to blackboard tools running on parallel computational systems.

Control Blackboards

In attempting to solve a domain problem, an AI system performs a series of problem-solving actions. Each action is triggered by data or previously generated solution elements, applies some knowledge source from the problem domain, and generates or modifies a solution element. At each point in the problem-solving process, several such actions may be possible. The control problem is: which of its potential actions should an AI system perform at each point in the problem-solving process?

Our approach to intelligent control problem-solving entails empowering AI systems to achieve the following behavioral goals:

- Make explicit control decisions to determine which problem-solving actions to perform at each point in the problem-solving process.
- Decide what actions to perform by reconciling independent decisions about actions that should be performed and actions that can be performed.
- Adopt variable grain-size control heuristics, including global strategies (e.g., first anchor all pieces of secondary structure in partial solutions; then refine the most credible partial solutions), local objectives (e.g., fill in gap *g* in the current solution), and general scheduling policies (e.g., exploit the most reliable knowledge sources).
- Adopt control heuristics that focus on whatever action attributes are useful in the current problem-solving situation, including attributes of their knowledge sources, triggering information, and solution contexts.
- Adopt, retain, and discard individual control heuristics in response to dynamic problem-solving situations.
- Decide how to integrate multiple control heuristics of varying importance.
- Dynamically plan, interrupt, resume, and terminate strategic sequences of actions.
- Reason about the relative priorities of domain and control actions.

In sum, systems following the proposed approach would forgo efforts to predetermine "complete" or "correct" control procedures that anticipate all important problem-solving situations. Instead, they would develop control plans incrementally while solving particular domain problems, adapting their behavior to a wide range of unanticipated problem-solving situations. (See [28] for more discussion.)

To realize these system behaviors, we are investigating a blackboard model of control in which control knowledge sources operate concurrently with domain knowledge sources to construct, modify, and execute explicit control plans out of modular control heuristics on a structured control blackboard. The control blackboard has the levels of abstraction defined and illustrated in Figure 9. Its solution intervals represent problem-solving time intervals.

```
Problem         Problem the system has decided to solve
                "Elucidate the structure of LAC-Repressor Headpiece"
Strategy        General sequential plan for solving the problem
                "Anchor all secondary structures; then refine all partial
                 solutions that anchor at least one Secondary element"
Focus           Local (temporary) problem-solving heuristics
                "Anchor all secondary structures"
Policy          Global (permanent) problem-solving heuristics
                "Perform actions that generate control heuristics"
o-Do-Set        Pending problem-solving activities
                "Anchor-Helix helix 1 to Secondary-Anchor4
                 Anchor-Helix helix 1 to Secondary-Anchor5
                 Refine-Partial-Solution anchored by Secondary-Anchor1"
Chosen-Action   Problem-solving activities scheduled to execute
                "Anchor-Helix helix 1 to Secondary-Anchor5"
```

Figure 9: Levels of BB1's Control Blackboard with Examples from PROTEAN

We also have developed a vocabulary and syntax for expressing heuristics, as illustrated in Figure 10. A simple scheduler, which selects both domain and control knowledge sources for execution, has no control knowledge of its own. Instead, it adapts its scheduling behavior to the control plan currently recorded on the control blackboard.

```
Name            Focus1
Goal            (Eq KS-Type 'Anchor)
Criterion       (for Each-Anchor in ($Find-All 'Solid '((Role 'Anchor)))
                 always ($Object 'Copies Each-Anchor))
Weight          8
Rationale       "Incorporate a copy of each Anchor into at least one
                 partial solution before deciding which partial solutions
                 to refine"
Creator         Chosen-Action 5
Source          Strategy1
Type            Strategic
Status          Operative
First-Cycle     6
Last-Cycle      20
```

Figure 10: An Example PROTEAN Heuristic at the Focus Level

In previous research [28], we developed the blackboard model of control and demonstrated its applicability to the control knowledge used in HEARSAY-II [16], HASP [55], and OPM [30]. We have implemented the control blackboard and several control knowledge sources arising from that research in the BB1 system. We are now using the model to organize control knowledge for the PROTEAN system. We propose to continue refining the model by assessing its applicability in different problem domains and by developing control knowledge sources that are useful for particular problem classes.

Explanation Systems For Control Blackboard Systems

During efforts to solve a domain problem, an AI system should explain its problem-solving behavior. It should justify actions in terms of the situations that trigger them, the knowledge they use, and the solution elements they generate. It should also show how actions fit into an overall line of reasoning, what specific control heuristics they satisfy, and what alternative actions were considered. These explanation capabilities are defining characteristics of intelligent problem-solving. They are also pragmatically desirable as debugging aids for system builders and as credibility checks for domain experts.

We propose to investigate explanation in the context of the blackboard control model and its explicit representation of a dynamic control plan:

- The current scheduling rule for choosing among feasible actions (e.g., "Schedule the highest priority action");
- The current integration rule for combining an action's ratings against multiple control heuristics to calculate its priority (e.g., "Compute each action's sum of weighted ratings against operative heuristics");
- The operative control heuristics (e.g., "First anchor all pieces of secondary structure in partial solutions; Then refine the most credible partial solutions." "Exploit the most reliable knowledge sources.");
- Each action's ratings (0-100) against operative heuristics.
- Each action's priority, computed by applying the integration rule to its ratings.

A preliminary explanation mechanism, implemented in the BB1 system, constructs stylized explanations such as the one shown in Figure 11.

We propose to continue this line of research to develop explanation mechanisms

```
I recommend KSAR 6
Should I Display/Explain/Go/Charge-Ahead/Override: E

KSAR 6: Knowledge Source: Anchor-Helix
        Trigger Event: (Add Solid2)
        Context: ((Anchor Solid1) (Anchoree Solid2))
Control Plan:
        Scheduling Rule: Highest Priority KSAR
        Integration Rule: Sum of Weighted Ratings
        Strategy1: Anchor-Then-Refine
        Rationale: Incorporate a copy of each Anchor
                   into at least one partial solution
                   before deciding which partial solutions
                   to refine
          Focus1: (Eq KS-Type 'Anchor)  Weight 8  Rating 100
        Policy2: (Eq To-BB 'Control)  Weight 10  Rating 0
Priority: 800
KSARs with the same Priority: KSAR 7, KSAR 8, KSAR 9
```

Figure 11: Example of Preliminary BB1 Explanation

appropriate for potentially much more complex control plans and to tailor information selection and presentation to the different interests of system builders and domain experts.

2. Constraint Satisfaction

GOALS

The long-term goal of this part of our research is to produce tools for constructing symbolic constraint satisfaction (SCS) programs, and to analyze and experiment with them to determine their strengths and limits.

The near-term objectives are to implement and experiment with an SCS program in resource management and to generalize it into a prototype SCS framework. We have selected resource management as a test-bed for this research because it involves constraints of different levels of detail and different degrees of "firmness," it involves using the same constraints in the context of somewhat different tasks, it involves time-dependent constraints (e.g., a previously committed resource may become available again in the future), and it involves a large amount of symbolic information that we, as resource managers, know intimately. This work intersects the research on using the blackboard model for constraint satisfaction problems, discussed in the previous section.

MOTIVATION

Reasoning about constraints is a ubiquitous problem with many facets. It occurs in many important problem-solving activities in which a solution is constructed from primitive elements but there are constraints on how those elements are put together. In DENDRAL [43], for example, there were *a priori* theoretical constraints on the *meaningful* constructs and *a posteriori* experimental constraints inferred from the data gathered for a specific problem. Both sets guided the hypothesis generator toward plausible solutions (and away from implausible ones). More recently, the R1 (or XCON) [47] program developed at CMU uses constraints of both types to put together a near-optimal configuration of computer components (including racks and wires). The *a priori* constraints constitute the "rules of the game" -- the components that may and may not be used together, for instance. The problem-specific constraints come from the description of the computer buyer's requirements, such as space available, memory required, and so forth.

Constraint satisfaction problems have not been as well-studied in AI as troubleshooting and diagnostic problems. There have not been, for example, successful generic frameworks developed in which constraint satisfaction systems can be built easily. For troubleshooting systems, on the other hand, several frameworks have been developed and successfully transferred to military and industrial installations. We believe that academic laboratories must intensify research on constraint satisfaction.

In a very large space of possible solutions, each constraint may be taken as a specification of a subset of solutions. In the abstract, then, successive constraints narrow the solution space to just those solutions that lie in the intersection of subspaces specified by all the constraints. This is a first-order model of constraint satisfaction that can, in principal, be applied with constraints of all forms.[1] However, the first-order model must be modified to accommodate several complexities:

- The languages in which constraints and solutions are expressed are not necessarily the same. Some reasoning process must translate from one to the other.

[1] Mathematical methods for constraint satisfaction, while appropriate for many problems, depend on constraints being expressed numerically with some precision. We are concerned here with problems for which mathematical methods are not appropriate.

- Qualitatively different kinds of constraints may apply to a single problem. The problem-solver must integrate them.
- The available constraints may be incomplete. The problem-solver must either characterize the "family" of solutions consistent with the available constraints or choose an arbitrary member of that family.
- The available constraints may be incompatible. The problem-solver must either decide to compromise some of the constraints or identify a dynamic solution that vacillates (in time or space) between states satisfying incompatible constraints.
- There is a potential combinatorial explosion of hypothesized solutions. The problem-solver must restrict search.
- The computational cost of applying individual constraints may be high. The problem-solver must manage these costs.
- Resources available for carrying out planned actions in the real world are constrained over time -- e.g., previously committed resources become available again after a time.

BACKGROUND

The management of resources is a critical part of most decision-making operations. There are often constraint satisfaction problems in which symbolic and numeric constraints interact at many stages in the decision- making process. Sometimes the constraints are expressed in terms of (a) the goal to be achieved, (b) intermediate goals or states, (c) resources available, or (d) the process itself.

A clear instantiation of this class of problems is the management of financial resources. Financial management encompasses the planning and initiation of new projects and the administration of awarded funding for on-going projects and operations. In most institutional settings, the accounting tools for collecting, recording, and reporting information about actual financial transactions in the performance of work (e.g., salary, procurement, and reimbursement expenditures) are well developed. Typically such systems are able to report monthly and cumulative expenses against a project budget; attempt to capture transactions in progress (completeness and accuracy depending on where a given transaction is in the bureaucratic pipeline when the monthly accounting is run); and help with report abstractions, trend projections, and the mechanics of plan calculations. Increasingly, the resulting information can be available to users in electronic form.

However, the tools for the more judgmental aspects of resource management, planning, and subsequent resource allocation, are much more primitive. The integration of the conceptual planning for work to be done with the financial planning, expenditure initiation, and control processes needed to actually carry out the work is mostly handled in the heads of individual project managers and administrators. It is these human experts who cumulate the working knowledge and experience of how to allocate financial resources to achieve work goals while satisfying the constraints imposed by funding terms and conditions and governing policies and procedures of the funding agency and parent institution. In a research laboratory, considerable specialized expertise develops for managing particular types of work under particular funding arrangements. For example, there are experts at managing contracts, or computing equipment purchases, or electronic assembly subcontracting, or hazardous material procurement, or a myriad of other activities confronting the performance of work objectives. Unfortunately, such expertise is almost never taught and it is acquired through experience involving trial and error and communication of lore from friends who have already had similar experiences. Frequently, there are wide variations in the ability of individual managers to properly administer these matters because of differing levels of experience and even degrees of caring about such managerial details in the face of the primary professional goals of the group.

Many project groups develop local administrative systems, many of them manual or adaptations of spread sheet software packages (e.g., VisiCalc), to facilitate management tasks. But these help only with the mechanical numerical aspects of management and do not assist in the judgmental matters involving optimal use of resources for work goals or satisfaction of policy and procedural constraints. These systems give little help in selecting and filling out appropriate forms for personnel, procurement, or other transactions. They do not provide intelligent interactive planning help that automatically relates, for example, personnel assignments in budgets with supporting expenditures like salaries, supplies, travel, telephone, and publication costs, appropriate to the work group involved. They do not provide catalog information for budgeting purchases of computing equipment, instrumentation, parts, or other discipline-related items. They do not advise on proper cost allocation and documentation relative to funding terms to assure that costs will be allowed. They do not help with planning expenditures among overlapping funding support so as to effectively achieve work goals within funding constraints. They do not help with the integration of institutional financial performance data with on-going plans, locating errors and reconciling the interface between locally recorded commitments and actual expenditures. And they do not provide the required flexible modes of information presentation such as tables and graphs, monthly details and plan exceptions, subproject detail or aggregation, or cross-project distributions.

Now clearly the above functions combine knowledge from many sources -- some factual and some experiential; constraints from many sources -- some numerical and many symbolic; and frequently no unique solution exists for a given planning problem. Spread sheet programs provide a useful interactive mode of calculating and displaying information but they only do part of the task of assisting with the managerial judgements involving symbolic knowledge and constraints. We have, under separate funding, begun work on a prototype system to utilize some of the techniques developed over the recent past for knowledge-based system design to further facilitate computer assistance in the task of budget planning and resource management.

In the longer term, this is one example of a broader class of complex constraint satisfaction problems. Other examples include space allocation, hospital scheduling and triage, interpreting Nuclear Magnetic Resonance data with other information to determine protein conformations, and system design. In studying the financial resource planning problem, we hope to gain more experience with this class of problems in the hope of developing more general problem formulation and problem solving tools for dealing with them.

RESEARCH PLAN

We propose to build a constraint satisfaction program that is (a) general across several types of problems and (b) useful within one or more specific management problems. The shortcomings of spreadsheet software packages mentioned above will be addressed in the context of the prototype object-oriented system already implemented.

The first system uses strictly numerical constraints to aid in constructing a research budget. It is able to access data stored offline about default values for budget items, such as salaries for individuals, cost of specific equipment, and the university overhead rate. It uses windows to display information rather than the more restrictive spreadsheet. Subsequent improvements will focus mostly on incorporating symbolic constraints in extensions that allow:

- defining forms
- filling out forms consistently
- integrating information from forms with budget information
- producing projections under different perspectives
- managing the flow of expenses over the life of a project

We will target our experimental systems for workstations with bitmapped displays to take advantage of powerful graphics tools which we believe will be necessary for an effective human interface. We will use the existing computing resources of the KSL for this work, including Xerox D-machines, Symbolics 3670's, or possibly Texas Instruments Explorers, while keeping a view for software portability to other workstations that will undoubtedly become available.

We expect to evolve the AI portion of the design carefully, based on requirements. Our view is that the system will start out by taking on some of the onerous manual tasks of financial plan development, with better interactive capabilities and being database driven. It will then become increasingly effective as an advisor for planning, leading ultimately to a more active role in plan formulation and review.

3. Knowledge Acquisition

GOALS

The long-term goal of this research is to develop robust machine learning programs that can be integrated with a variety of intelligent systems, and to develop a set of criteria under which machine learning techniques can be successfully applied to different problem-solving architectures.

In the near-term, we propose to design, implement, and experiment with learning methods in different problem-solving environments. In particular, we propose to: (a) extend the work on induction with rule-based systems in the BB1 and HERACLES architectures; (b) develop methods for learning control heuristics in the blackboard architecture; (c) develop programs for learning by chunking (as already implemented in the subgoaling architecture of SOAR) for the classification architecture of HERACLES and the blackboard architecture of BB1. We also propose to extend our analysis of issues in building machine learning systems, specifically the role of noise, the role of examples, and the role of knowledge representation in machine learning.

MOTIVATION

Over the last decade, many machine learning programs have been implemented for special-purpose acquisition of new knowledge. They have been constructed with an eye to generality but with the generality lying mostly in the descriptions of ideas, not in the details of the method and certainly not in the code. The details need to be analyzed so that the strengths and limits of different methods can be assessed in different contexts.

Domain-independent methods are limited by their lack of semantics underlying the names of features being manipulated. Statistical methods, for example, are generally applicable (for data described with numerical features) but lack the ability to use specialized knowledge of a domain that could increase their power. The tradeoffs between generality and specificity in machine learning systems need to be analyzed in order to build powerful learning methods that apply to more than single tasks. Meta-DENDRAL [43], for example, was completed in our laboratory about 1979, but was not developed outside its original task area until 1985 [19].

In the future, it is imperative that methods for machine learning be well enough understood that "off-the-shelf" packages can be constructed and made available for the different classes of intelligent systems we now know how to build. For example, diagnosis and troubleshooting problems are modestly well understood. There are framework systems, like EMYCIN and its commercial cousins, that aid in the construction of a new expert system, e.g., a diagnostic problem solver for a specific task.[1] But there are no pre-packaged learning programs that can be added to the resulting expert system to give it the ability to learn. Since learning takes many forms, there is not just one single package that will serve all purposes. If there is a large library of cases, then learning by induction may be a good way to begin building, or to refine, a knowledge base. If a problem solver is in routine use, then it may be more appropriate to couple it to a learning program that will refine the knowledge base by interacting with specialists using the system, or by watching -- and forming a model of -- what they do.

BACKGROUND

[1] The classes of problem solving systems, themselves, need to be better characterized so that framework systems like EMYCIN can be reliably matched to proposed problem areas. Some work along those lines has been undertaken, on which we will build [10, 5].

Recent work indicates the feasibility of building domain-independent learning programs that use knowledge supplied from the outside to guide the learning. Several overview articles written by members of the KSL and others summarize and analyze the state of machine learning and knowledge acquisition systems. Among the most influential of these on our own work was the "Models of Learning Systems" paper in which learning was viewed as a problem-solving activity with distinct components. It is shown in Figure 12 below.

```
Instance                    Performance
Selector                    Program
        \                  /
         \                /
           Blackboard
          /              \
         /                \
Editor                      Critic
```

Figure 12: The components of a Learning System.

The problem-solving vocabulary, assumptions, and procedures are defined for all of the components of the system within a world model. One component, the instance selector, chooses training instances to present to the performance program. Performance is critiqued by the critic, whose advice is implemented by the learning element. These steps are not always separate or all automatic.

In the last several years, we have undertaken several experiments in machine learning. Most of these are implemented programs either completed or near completion. Most of these have been done on SUMEX using a biomedically relevant task area as a test domain. They are briefly described in this section with some of the conclusions that are emerging from preliminary analyses.

- *INDUCTION -- LEARNING FROM EXAMPLES*
  - Meta-DENDRAL -- a model-driven induction program that learned new inference rules for the DENDRAL program. It demonstrated the power of heuristic search as an induction method, the power of a "half-order theory" for constraining the search, and the power of a two-tiered search strategy with approximate search followed by detailed search. Its primary mode of learning was generalization from examples, with specialization added in a separate, final step.
  - Version Spaces -- a bidirectional search program that also learned new inference rules for DENDRAL. It demonstrated the power of using generalization and specialization together to refine a subspace of allowable rules (or concept definitions).
  - PRE -- a program that uses a partially formed theory to interpret data in the context of learning refinements and extensions of the partial theory. This "theory-driven data interpretation" program uses constraint propagation methods to keep track of interrelationships in the emerging theory.
  - JAUNDICE -- an inductive learning program that learns new rules for performance programs written in EMYCIN by generalizing and specializing from cases in a data base. It demonstrates the power of bidirectional search, the power of reducing the number of features and filtering out noise.
  - PIXIE -- a program that learns a model of a student's behavior in a

tutoring context from a record of correctly and incorrectly solved problems. It shows the power of starting with a model that "should" produce correct I/O pairs and systematically perturbing the model until the predicted I/O matches the observed data.

- *KNOWLEDGE ENGINEERING -- LEARNING FROM EXPERTS*

  - DENDRAL -- the activity of knowledge engineering was first described (but not named) in 1971 [7] in the context of DENDRAL. It was recognized there as a bottleneck in building knowledge-based programs using experts as sources of knowledge.

  - MYCIN -- several of the now-classical difficulties of knowledge engineering -- such as the problems of welding consensus from incompatible knowledge sources and maintaining a consistent KB -- were first described in the context of our work on MYCIN.

  - TEIRESIAS -- a program that used meta-knowledge in interactively debugging and maintaining a KB (specifically MYCIN's KB). This work demonstrated the value of explanations for understanding the contents of a KB and the value of meta-level knowledge for helping edit a KB efficiently and consistently.

  - EMYCIN -- a generalized framework for building MYCIN-like consultation systems. It incorporated an abbreviated rule language (ARL) that allows an expert on knowledge engineering to write new rules in a stylized form that is easier than LISP (but more telegraphic than English).

  - ROGET -- an experimental expert system whose domain of expertise is knowledge engineering. Although never used outside our laboratory, it showed the extent to which our own knowledge about knowledge engineering could be codified.

  - MOLGEN -- within the UNITS package; MOLGEN included a KB editor that experts, not knowledge engineers, use to maintain a large, complex KB. It demonstrated that experts can and will learn a powerful, but syntactically simple, KB editor when the benefits outweigh the costs.

  - BLUEBOX -- an EMYCIN system with considerable expertise gleaned from the literature by students. It showed that an expert system can be built without tying up an expert if the domain is well structured and well agreed-upon.

  - OPAL -- an interactive KB editor still under construction. It shows the power of building knowledge structures on top of a well designed language. In this case, the language is one of procedures, with temporal predicates.

- *LEARNING BY WATCHING*

- ODYSSEUS -- a program nearing completion that learns by mapping what it infers an expert knows (by watching what an expert does) onto a KB for an expert system. It demonstrates the power of using a modelling system (originally constructed for modelling a student in an intelligent tutoring system, GUIDON) to determine the rules an expert probably uses, without asking the expert directly.

- *LEARNING BY ANALOGY*

- NLAG -- a program that *uses* an analogy, stated as a simple hint, "b is like a", in order to construct new rules in domain B from a KB already built for domain A. It demonstrates the power of an abstraction hierarchy for relating concepts in similar domains and for mapping from one set to another.

- *LEARNING FROM THE LITERATURE*

  - REFEREE -- a prototype EMYCIN program that reasons about the contents of journal articles in order to find new rules in those articles. (Note that answers to questions are supplied by a student who reads the articles, not by a program, or an expert, who reads the articles.) Preliminary results indicate that some journal articles are written clearly enough that a program with only general knowledge of the domain can guide a novice to the new knowledge contained in them.

  - BLUEBOX -- (see above). One lesson is that the literature of a well structured domain can be interpreted correctly by novices to build the KB for an expert system.

- *LEARNING FROM EXPERIENCE*

  - DENDRAL -- a dictionary of previously solved subproblems increased the efficiency of DENDRAL's heuristic search. It illustrated the power of rote learning but also pointed out clearly the tradeoffs between storing and recomputing answers.

  - AM/EURISKO -- programs that use previously computed material to aid in the discovery of new knowledge. These programs illustrate the power of combining existing elements in a KB in various interesting ways in order to construct new elements that are interesting and useful.

  - SOAR -- a general problem-solving system under construction that incorporates a methodology for "chunking", i.e., rote learning with generalization. Preliminary results point to chunking as an effective method for learning from experience in a broad class of problem solving systems.

- *STATISTICAL METHODS*

  - RADIX -- a program that finds statistical correlations in a very large data base, and then discovers whether or not the empirical association is semantically interesting.

RESEARCH PLAN

A) Induction

We propose analyzing the strengths and limits of the generalization and specialization methods in the JAUNDICE program [19], mentioned above, and to implement the same methods in the HERACLES and BB1 architectures. As developed, those methods can be used to learn rules of an EMYCIN syntax from case libraries. The primary techniques are successive specialization guided by general knowledge of the domain, and successive generalization guided by positive and negative examples in the case library. The specialization and generalization operators, as written, are closely tied to the rule-based formalism, but will be recast to work with the slot-attribute representation used in BB1 and HERACLES.

As described in [19], inductive learning can be considered as either or both of top-down specialization of a general concept or bottom-up generalization of the descriptions of specific instances. The rules used in the JAUNDICE system, which we propose to implement in other systems are summarized in the table below.

```
Rules of generalization:   1. Dropping conditions
                           2. Climbing up the value hierarchy tree
                           3. Creating new symbols
                           4. Taking minimum or taking maximum
                           5. Allowing disjunction

Rules of specialization:   1. Adding conditions
                           2. Climbing down the value hierarchy tree
                           3. Closing interval
```

Figure 13: Summary of Rules of Generalization and Specialization by Fu [19]

The search proceeds stepwise using the heuristic rules summarized above as plausible "move generators" in the space of rules, and checking alternative formulations against the data in the case library, as in Meta-DENDRAL [43].

The methods developed by Fu & Buchanan in the context of EMYCIN systems, will be generalized so that the dependence on a rule-based representation of knowledge will be removed. This requires clean separation of the credit assignment methods and the editing methods, as discussed in [8]. The credit assignment programs need to determine generally what is wrong and what to fix (when predictions are false), and then communicate this information to the editor in a high-level, representation-independent, language which the editor translates into specific changes for the knowledge structures being used. In a rule-based representation, for example, inferential links are represented exclusively as premise-action pairs of conditional rules. In a frame-based system the inheritance links carry some of the same kind of inferential information. Thus the editor needs to know the semantics, as well as the syntax, of slots and attributes in order to change the appropriate constructs.

B) Learning Control Heuristics by Experience

Articulating and coding domain knowledge is time-consuming for both the domain expert and the knowledge engineer. Acquiring control knowledge poses additional problems [22], [25]. Control knowledge appears to be more difficult for experts to retrieve than domain knowledge and they have difficulty distinguishing domain and control knowledge. Experts produce general heuristics during questioning, but use more specific heuristics during problem-solving. Stimulating experts' retrieval of a comprehensive set of heuristics may require analysis of many example problems that produce no new domain knowledge. At the same time, powerful control knowledge is essential for the solution of many problems.

We propose to study automatic learning of control knowledge in the context of BB1. As discussed above, all cognitive activities in BB1 systems are performed by knowledge sources that are triggered by changes to objects on the blackboard and, when executed, produce new changes to objects on the blackboard. These include domain knowledge sources that construct solutions to the domain problem on the domain blackboard and control knowledge sources that construct control plans for the problem-solving process on the control blackboard [28]. Similarly, knowledge sources for learning will introduce new control heuristics into the current control plan and they will construct new control knowledge sources to generate the new heuristics in the future.

We envision a range of potential learning knowledge sources, including some that learn new control heuristics, some that learn more general or more specific forms of known heuristics, and some that expand or restrict the applicability of known heuristics. Within each category, some learning knowledge sources simply replace the knowledge engineer and interact directly with domain experts. For example, the knowledge source Understand-Preference, a prototype version of which we have already implemented [29], is triggered when a domain expert overrides BB1's scheduling recommendation. Its action interacts with the expert to determine the reason for the override and encodes a corresponding new heuristic. Other learning knowledge sources could operate autonomously. For example, the knowledge source Attribute-Results might be triggered by dramatic improvement (or deterioration) in the current solution to the domain problem. Its action would attribute the change in solution rating to preceding actions and encode a heuristic favoring such actions. Evaluate-Heuristic, another autonomous knowledge source, might be triggered when a new control knowledge source is executed. Its action would evaluate subsequent changes in solution rating and adjust the posted heuristic's assumed importance (Weight) accordingly.

The proposed work will develop specialized mechanisms for these different kinds of learning. For example, Understand-Preference compares attributes of the action recommended by the scheduler to corresponding attributes of the action preferred by the expert and, with the expert's assistance, diagnoses the key differences. By contrast, the knowledge source Evaluate-Heuristic requires a mechanism for measuring and evaluating changes in the quality of a solution and for distributing "credit" for those changes among simultaneously active control heuristics.

BB1 provides a rich and well-structured foundation for learning in its explicit, structured representations of all blackboard objects, knowledge sources, and potential actions. The structure and semantics of BB1's control blackboard entail a prototypical form for all control heuristics used by the scheduler:

```
Goal:       (Function <KSAR:Attributes> <Other-Arguments>) = {0-100}
Weight:     {1-10}
Criterion:  (Predicate) = T/F.
```

A heuristic's Goal is a function that, when evaluated for a potential action, produces a rating 0-100. Its Weight is a number 0-10 that signifies the importance of an action's rating on the Goal function. Its Criterion is a predicate specifying an expiration condition that, when met, signifies that the Goal is no longer desirable. All learning knowledge sources will attempt to construct (or modify) control heuristics in this prototypical form. They also will attempt to construct control knowledge sources whose triggering conditions describe appropriate situations in which to adopt new heuristics and whose actions post the new heuristics on the control blackboard.

The proposed work will supplement the BB1 foundation with additional knowledge of canonical forms for semantic classes of control heuristics. For example, control heuristics that rate actions on attributes with numerical values might incorporate Goals in the canonical form:

```
(Translate-Value-To-Scale KSAR:Attribute Maximum-Value),
```

in which observed values on the target attribute are translated into corresponding values on the required 0-100 scale. Alternatively, they might incorporate Goals in the canonical form:

```
(Compare-To-Threshold KSAR:Attribute Threshold),
```

in which observed values on the target attribute are rated 100 if they are above some threshold, and 0 otherwise. Obviously, there are many alternative canonical forms that are potentially appropriate for attributes with different data types (e.g., numerical, literal, list). Learning knowledge sources must determine which form is appropriate for each new heuristic.

Although we will develop and evaluate learning knowledge sources in the context of the PROTEAN system for protein structure analysis, the knowledge sources themselves will embody generic learning mechanisms applicable to a wide range of problem domains. We will incorporate these learning knowledge sources in the BB1 environment.

C) Knowledge Engineering

We propose to build interactive aids for knowledge engineers in the context of the BB1 and HERACLES frameworks. Many of the aids in EMYCIN, although developed nearly a decade ago, have never been duplicated, or have only been partially duplicated, in other contexts.

These ideas include:

1. meta-level constructs to guide the acquisition and checking of new knowledge;
2. interactive debugging aids for tracking down the source of an error in the context of an incorrect conclusion;
3. explanation facilities.

HERACLES is a tool for building expert systems that we have generalized from our experience with NEOMYCIN, a program designed to clarify the knowledge structures and reasoning processes of MYCIN. HERACLES solves problems by classifying them in terms of a set of pre-enumerated solutions, a method we call *heuristic classification*. For example, a generic form of heuristic classification, commonly used for solving diagnostic problems, is *causal process classification*. We have been studying how causal processes are classified in medical diagnosis, and have recently applied our model to the problem of diagnosing surface flaws in cast iron.

In causal process classification, data are generally observed malfunctions of some device or process, and solutions, pre-enumerated in the program, are abnormal processes causing the observed symptoms. We say that the inferred model of the device, the diagnosis, *explains* the symptoms. Only the simplest devices and processes, can be adequately described in terms of function/structure models, enabling a principled comparison of faulty behavior to intended design. Instead, it is necessary to construct a *causal network* that relates normal and abnormal states to observed behavior and ultimate fault etiologies.

While causal networks of this sort have been incorporated in medical diagnostic programs, for example, for more than a decade, the principles by which they should be constructed is still an area of research. In our own work, we have been investigating heuristics for constructing such networks in knowledge acquisition dialogues. We have discovered that an expert's terminology and explanations of causal processes must be carefully analyzed for the resulting network to be coherent and applicable to many problems. For example, an expert may say, "a brain-tumor causes a brain-mass-lesion." But a network simply linking these two terms will be meaningless: a brain-tumor is *a kind of* brain-mass which *causes* a brain-lesion (cut). The two terms cannot be linked simply by either cause or subtype because the term "brain-mass-lesion" bundles together a location, a cause, and an effect.

In our ongoing research, we propose to continue this kind of analysis to develop a program that can help a knowledge engineer construct a principled causal network. We believe that a promising approach is to enrich the representational structure of our network language, so that the program knows not only that "X causes Y", but also has enough detailed knowledge so that it can explain why the connection is plausible. Such a program could aid the knowledge acquisition process by automatically critiquing the evolving network. Moreover, the program would ask questions to help it fill in the gaps and lack of coherency it detects.

Using the above example, after being told an implication (ordinary heuristic rule) relating brain-mass-lesion and brain-tumor, the program would attempt to classify these terms as processes or substances, note the locations, and isolate the particular causal interaction (mass causes a lesion). The key to such a capability is a representation language that defines concepts in terms of a relatively small number of relations (such as the conceptual dependency notation of Schank), plus generic knowledge of physical processes (e.g., the idea of a mass growing in size severing an enclosing substance). A great deal of research in qualitative reasoning of physical processes [3], particularly the research of Wendy Lehnert, lays the foundation for this kind of investigation.

The learning program we will construct could be termed "the advice requester." We believe that the ability to ask good questions is the mark of a good student or researcher, and it can greatly focus the learning process. Asking good questions requires relevant background knowledge, so the learner can learn something new by relating it to some facts or some general framework he already understands. This process can be complex, because there are levels and perspectives for understanding. What may at first appear consistent, could become puzzling later as new gaps appear in an evolving network. Concepts in fact change their meaning as exceptions and complex special cases come to light.

Learning by asking is a form of knowledge-intensive learning, to be contrasted with research in automatic learning (becoming more efficient). For knowledge engineering, such an approach is a dramatic switch from giving the program surface causal rules that it in no sense understands, to giving a program knowledge of underlying causal models that enable the program to justify its causal network. Most importantly, these models provide a set of expectations of states and faults that might be included in a causal network.

To take an example from another domain in which we are working, iron casting, one fault is a shrinkage cavity. Generic knowledge would indicate that a cavity is an absence of material, and that for casting the source of material is what is poured and a reservoir (part of the mold) to allow for shrinking. A built-in generic model would indicate three reasons why a source of material does not arrive at the sink: insufficient supply (reservoir is too small), supply lost by leaking, and blocked flow from source to sink. These three generic causes set up expectations for specific causal processes that will appear in the state network. A given knowledge base might refer to a model only once, but a library of such models would form the basis of a powerful knowledge acquisition program that could learn about new domains fairly quickly. We believe that this generic library of processes is part of what we call common sense knowledge.

An advice requester that would be as proficient as our best knowledge engineers is obviously not going to be constructed in a year or two. Our approach will be first to study the causal networks we have constructed in medicine and casting, and re-represent the knowledge in structures that include the generic, underlying abnormal processes. Next, using a method we have found to be advantageous in the past for refining a knowledge representation, we will construct a simple teaching program that can explain such a causal network and help the student critique an incomplete network. Ultimately, we believe that teaching students to think like knowledge engineers, that is *to learn the process of asking good questions*, may be even more valuable than directly trying to convey our products, the constructed knowledge bases.

4. Qualitative Simulation

GOALS

In the context of the Molgen-II project, we are exploring the process of scientific theory formation and modification by computer. Qualitative simulation of biological processes is an important part of this goal because it is necessary to ask about the results of hypothetical experiments in the course of theory formation and running a detailed simulation is often too expensive.

MOTIVATION

We are carrying out this research by studying a specific biological system: the regulatory genetics of the *E. Coli* tryptophan operon (the *trp system*). In the mid 1960's Dr. Charles Yanofsky (who is a collaborator with us on this project) began to probe the existing theory of gene regulation in this operon. Yanofsky's initial experiments revealed a number of anomalies. Since that time, Yanofsky's research (which continues today) resulted in the discovery of a totally new mechanism of prokaryotic gene regulation, and continues to refine our knowledge of exactly how this mechanism functions.

Our goal is to build a machine learning system which will accept an initial theory of gene regulation equivalent to that which Yanofsky began to probe in the 60's. We will then present our system with a series of experimental results based on Yanofsky's early observations. The learning system will then propose, implement, and attempt to confirm possible modifications to its theory of gene regulation.

We view *theories* - such as that of the trp operon's function - as problem solvers. The inputs to these problem solvers are descriptions of hypothetical experiments. The problem solver's outputs are descriptions of the predicted results of these experiments. Thus our learning program will be attempting to improve the predictive performance of a problem solver in bacterial regulatory genetics.

This research in machine learning presumes the existence of a simulator of the trp system. Building such a problem solver in itself raises interesting AI research issues in qualitative simulation. And building such a system in a form which can be *reasoned about* by another program (the learning element) complicates the problem even further.

Below we discuss our past work on the construction of two versions of such a problem solver ("the simulator"). We then outline a number of interesting research issues which this work has raised, and the approaches we plan to pursue in the construction of the simulator.

BACKGROUND

Version I

An exploratory version of the system was built in the Spring of 1984. The system was constructed using the UNITS system - one of the first general-purpose expert system building tools.

This first system was more of a success as a static knowledge base than as a dynamic simulator. Building this system forced us to come up with a concrete conceptualization of the problem domain: we determined the full range of objects the system would have to simulate, and considered what types of properties and internal states these objects have, and how they should be represented within the UNITS system. This knowledge base was examined several times by our biologist collaborators (Yanofsky and Dr. Robert Landick - a post-doctoral fellow in Yanofsky's lab) to help us detect errors and omissions.

The first system never contained much simulation capability. We did provide a mechanism whereby the state of the transcription mechanism could be determined after the user specified experimental conditions such as approximate tryptophan concentration and whether or not various objects such as the trp-R repressor and the trp promotor contained deleterious mutations or not. The simulation capability was essentially provided by backward chaining on the slot values of relevant units, with the actual inferences carried out by Lisp code attached to some slots.

We learned a number of things from this prototype system. The knowledge base we created became a concrete record of the objects relevant to problem solving in this domain, and of design decisions regarding their representations. We also discovered a number of things about the UNITS system:

1. Its knowledge base editor ran fairly slowly
2. We encountered and fixed several significant bugs
3. Its rule language is fairly awkward
4. Its inheritance hierarchy lacked some important features, such as the ability of a given object to inherit slots from more than one parent class.

(Note that points 1 and 2 result from UNITS having been developed and maintained within a university research environment.)

We also confirmed an observation made long ago by other AI researchers. Previous work has shown that the simpler a language is, the more amenable it is to being both executed by one entity and interpreted by another entity (such as an explanation facility). This is one reason expert systems are now often encoded in production rules rather than Lisp. It became quite obvious that if our learning element is forced to reason about a simulator containing Lisp procedures, it would be significantly more complex than if the simulator were written in another language. Simple as the syntax of Lisp is, even a reasonable subset of full Interlisp would contain quite a large number of fairly complex constructs, and would complicate the learning element tremendously.

We also made an interesting observation about how building an expert system can help experts think about their own domain. We will consider two examples of this particular idea. Both involve subclass units which were defined in the knowledge base by Karp and then discussed with Yanofsky and Landick. One subclass was called "DNA Segments" and was intended to include contiguous segments of DNA with discrete functions, such as: promoters, terminators, genes, and operators. Among the properties associated with this class were: sequence, position within some larger functional piece of DNA, and "generalized sequence" - an attempt to capture those sequence elements common to a given subclass of DNA Segments such as promoters. The other defined class of interest was termed "Molecular Switches". This was an attempt to represent the general notion of a molecule with two functional states, where transitions between states are caused by the binding and dissociation of the molecule from some other molecule. Examples of Molecular Switches are operators, promoters, and repressors.

In both cases Yanofsky and Landick expressed interest in these concepts, and noted that biologists had coined no terms for them. This suggests that these concepts are in some sense new to biologists. We hypothesize that the process of constructing an expert system will naturally lead to the identification of such general concepts - or, equivalently - to the creation of analogies between known concepts.

The reason for this is that in attempting to represent the behaviors of N different entities, it is often much more efficient (with respect to development time and code volume) to develop one general-purpose procedure which yields the N different behaviors given different parameter bindings, than it is to develop a different procedure for all N cases. It is the knowledge engineer's job to search for such general procedures.

Version II

Recently we have begun building the next version of the simulation system. We are implementing this using the KEE knowledge engineering tool developed by IntelliCorp. This will free us from all the limitations of the UNITS system mentioned above. We have accomplished the initial obvious goal of porting the knowledge base defined using UNITS to KEE.

Related Work

Recently a significant amount of work has been done in AI in Qualitative Simulation (de Kleer and Brown, Forbus, Patil, Kuipers). While this work is somewhat relevant to the research we propose, there are several reasons why it is not sufficient.

First, most of this work attempts to simulate systems described by *Physics* using *differential equations*. Much of this work is an attempt to generalize numerical differential equations into *qualitative differential equations*. However, Biology is a much younger science than Physics, and as such does not describe its mechanisms to nearly such a quantitative degree. Differential equations are rarely if ever used by Molecular Biologists, and hence qualitative differential equations do not

RESEARCH PLAN

The next step is to define the behavior for these objects so that actual simulations can be executed. This raises the question: in what language should this behavior be defined?

We rule out Lisp for reasons discussed earlier. We also believe production rules are not a good language for defining this behavior, for reasons that will be outlined below. We now discuss the features we believe the simulator should provide, describe research questions these features raise, and consider what constraints such a simulator imposes on an underlying implementation language.

Reasoning At Varying Levels Of Detail

We believe it is important that the simulator be able to reason at varying levels of detail depending upon the demands of a particular problem. That is, it should be possible for the simulator to solve many problems without simulating every single process it knows about in the most detailed manner possible. Rather, given a problem statement the simulator should perform meta-level reasoning to determine which processes to simulate, and at which of several possible abstraction levels to simulate each process. For example, in an experiment involving an otherwise normal *E. Coli* cell with a deleterious mutation in its trp-R protein, it should not be necessary to simulate the RNA-synthesis actions of RNA-polymerase at the nucleotide level. A more abstract representation of this process can be used (e.g., at the DNA Segment level).

It should be obvious that humans solve problems in this way as illustrated by the preceding example (that is, biologists can predict the outcome of this experiment correctly without employing such a detailed simulation). As human performance in this domain is reasonably high, there is reason to believe that this approach is not a bad idea. But what reason do we have to believe it is a good idea? Why not build a simple simulator that executes at one constant level of detail and be done with it?

This simulator is really only a sub-system of the whole discovery system, and as such could be called on many times during a given "discovery deliberation". It is thus quite possible that the speed of the simulator will affect the tractability of the discovery problem.

In addition, learning itself is usually subject to large combinatorial explosions. Consider learning to be a search through a space of concept descriptions, where generalization and specialization are among the state transformation operators. The more concept description primitives there are to combine, the less feasible this computation becomes. If the simulator represents object structure and function at one very detailed level, there will be a huge number of primitives to recombine. But if objects are represented at different levels of abstraction, learning too may proceed using "primitives" at higher levels, where presumably there are few primitives at the less detailed levels.

In Biology and the other Natural Sciences, many discoveries consist of the addition of detail to some model. Objects (e.g., ribosomes, atomic nuclei) which were once considered to be primitive black boxes have their insides probed to reveal a complex inner structure, or the range of their observed behaviors may increase. If our simulator is designed to represent and execute theories at different levels of detail, adding detail to an actual theory could be as natural as adding a new cell to the front of a linked list.

Another issue is user interaction. Users will want to include high level vocabulary terms in their specifications of experiments. And similarly, they will want to see these terms used in predictions. (Note this constraint does not force the system to be able to reason at *varying* levels of detail).

The issue of reasoning at different levels of detail is very relevant to current research in expert systems regarding "Deep vs Shallow reasoning". Some researchers argue that the "shallow reasoning" or reasoning from "empirical associations" used by traditional expert systems implemented in production rules (e.g., MYCIN) is qualitatively different from "deep reasoning" or reasoning from "first principles" which human experts are able to use when their "shallow reasoning" fails, or when "deep" explanations are required. I claim that while it is certainly important to be able to reason in a more detailed manner when a standard approach to solving a problem fails, and that it is crucial to be able to provide deeper justification for a line of reasoning than simply citing rules X and Y, that there is no absolute distinction between "deep" and "shallow" reasoning. What is possible is to distinguish one line of reasoning from a *deeper* line which justifies it. The construction of this simulator should help to prove this point.

Production rules have not been designed for the task of reasoning at varying levels of detail. It is important to design a language which explicitly provides this ability.

Knowledge Representation

The initial work done on the simulator has alerted us to unresolved issues in knowledge representation related to inheritance hierarchies. The inheritance hierarchies of both UNITS and KEE provide the ability to define properties of a given class unit which are inherited by subclasses or members of that class. But in fact this notion of class partitioning blurs together - and is used by knowledge engineers to represent - at least four different concepts. These are the concepts of *class*, *abstraction*, *prototype*, and *object decomposition*. Inheritance hierarchies also force one to make some choice about what is a *primitive* object in a given domain. Yet the notion of an individual is a difficult concept to define - philosophers have devoted entire books to it. AI could benefit from a systematic study of all five of these concepts, and this simulator provides a challenging context in which to study them.

Another idea to explore is object behavior structuring. A given object may potentially exhibit several different behaviors. For example, messenger-RNA binds to different molecules, is translated into protein, and is slowly degraded within the cell. Consider two different approaches to representing this behavior. In an object oriented approach, all behavior specifications for a given object are viewed as part of that object. Thus, at a given instant in time it is easy to determine exactly what behaviors a given object will demonstrate. Consider a process-oriented structuring of behavior. Using this approach, a given behavior is structured within some larger process of which it is a part. Thus, the binding of mRNA to a ribosome would be viewed as one element of the complex process of translation, which would be considered quite distinct from the process of mRNA degradation. This makes it difficult to reason about sets of asynchronous processes operating in parallel, but provides an easier way of reasoning about a long series of events which are causally connected.

It is not clear what the precise trade-offs between these two approaches are. It may sometimes be necessary to employ both, which would probably require translation between the two. This distinction has been explored by the Computer Systems community, but these ideas should be transferred to the AI community and would probably gain some clarity in the process. seem to be useful simulation tools.

Second, the other work in qualitative simulation simply has not addressed many of the issues we propose above, such as reasoning at varying levels of detail and making more sense out of inheritance hierarchies.

Summary

We propose the following:

- To design a process specification language which will form the heart of the simulator for the trp system. This language will be fairly similar to production rules, but will overcome the shortcomings of production rules as discussed above.

- To implement an interpreter for this language which will allow both forward simulation to predict the results of a specified experiment, and backward simulation, to suggest experiments which would explain an observed result.

- To implement an actual simulator for the trp system.

- To explore possible means by which the simulator should decide at what level of detail a simulation should be run to solve a given problem.

- To explore issues in knowledge representation concerning the concepts of an abstraction, a prototype, a class, a composite object, and an individual.

5. Additional Basic Research of the Knowledge Systems Laboratory

In addition to the core research described above, there is considerably more research in the KSL that draws on the SUMEX resource and that inter-relates to the whole SUMEX community. This is briefly summarized below in three main projects of the HPP, LOGIC, and HELIX groups of the KSL. (See Appendix A on page 285 for a description of the organization of the KSL.)

Research on Multiprocessor Architectures for Symbolic Computation

As the aspirations for applied AI work rise, expert systems are becoming more complex, and the symbolic computations involved more compute-intensive. Medical and biological applications share the widely felt need for more processing per dollar in the future.

VLSI technology, of course, offers the prospect of inexpensive high speed computing, but only if methods can be found to organize large collections of processors and memories in systems for concurrent (parallel) processing. The Heuristic Programming Project began work on this problem in the mid-1970's, with SUMEX computer support, in a project called HYDROID, whose major result was a system for a network of processors known as Contract Net [67]. HYDROID was reborn in 1983 as Advanced AI Architectures (AAIA), and has received funding support from DARPA and computing support partially from SUMEX.

In the AAIA project, the proposed architectures are studied in simulation (on Symbolics workstations). The underlying architecture is a distributed processor and distributed memory network, simulated with our CARE simulator. On top of CARE various experiments in the development of Concurrent LISP are being done. Above the LISP level are levels of knowledge access and problem-solving framework. At the knowledge level, methods are being studied for rapid retrieval of objects and rules in a multiprocessor net. At the problem solving level, we are studying the "parallelization" of the Blackboard framework. The Blackboard framework was chosen because we felt that, overall, it was the most powerful of the modern AI problem solving organizations and offered significant opportunities for the exploitation of parallel processing.

The top level is the level at which applications are programmed, and the opportunities for parallelism at this level are mostly domain- dependent. However we are studying in detail applications of the particular class known as signal-understanding (or signal-to-symbol transformations), hoping to discover a few generalizations applicable to the class.

If the levels are "factored" carefully and correctly, the speed-ups from parallel processing,each level to the next, will multiply (!), yielding overall a major system-wide speed-up from modest gains at each level (which is all that one can hope for at present). The goal of the AAIA project is to refine the level-factoring and the speed-ups at each level over the next 2-4 years to produce an overall gain from multiprocessor "parallelism" of at least one hundred times that of conventional serial machines (as measured by the simulator).

A Retrospective of the AGE Experiments

The scientific work of the KSL is largely experimental in nature. Ideas are embodied in software systems and are tested in significant applications. The AGE project was one of those lengthy experiments. From the beginning it was supported by SUMEX as core research. It had multiple goals: a) to provide a readily useable software package for developing expert systems employing the Blackboard framework b) to study the Blackboard framework itself with a view toward simplifying and generalizing its various mechanisms and c) to study the problem of how to build a "knowledge engineering workstation" environment (i.e. put KE expertise into the box).

AGE-1 exists, has been widely used, and is widely distributed. Many technical reports and papers exist. At the KSL, the scientific tradition is to bring together, summarize, and interpret the results of our multi-year thematic studies in a single scientific monograph that represents the best scientific sense we can make of the many experiments in the line of study. We did it with DENDRAL (Lindsay, Buchanan, et.al.), later with MYCIN (Buchanan and Shortliffe). We will soon begin the effort to do the necessary and appropriate AGE retrospective study. It will be done as a "background" effort to other activities and will take about three years (elapsed time).

Research on Logic-Based Systems and Systems with Self-Awareness

One of the key limitations on the technology of logic programming is that the usual logical rules of inference are too weak. While traditional logical implication is an essential part of expert reasoning, by itself it is inadequate to explain the cognitive performance of human experts or to serve as the sole basis for a practical logic programming technology. Over the next five years we propose to study and implement four specific advanced reasoning techniques, viz. uncertain reasoning for resolution, theory formation based on measures of probability and simplicity, efficiency-enhancing theory reformulation, and counterfactual implication.

The key idea underlying logic programming is that of programming by description. In traditional software engineering, one builds a program by specifying the operations to be performed in solving a problem, i.e., by saying HOW the problem is to be solved. The assumptions on which the program is based are usually left implicit. In logic programming, one constructs a program by describing its application area, i.e., by saying WHAT is true. One makes one's assumptions explicit and leaves implicit the choice of operations.

Uncertain Reasoning

The actual techniques used to implement uncertain reasoning facilities have increased in sophistication since the introduction of "certainty factors" in MYCIN; the approach which has received the most attention recently is the use of Dempster-Shafer theory [64]. Here, ranges of probabilities are considered instead of specific values; this has the advantage that it is possible to describe situations where one is uncertain as to the accuracy of one's information by representing it using a wide interval of possible probabilities.

Existing work at Stanford has laid a theoretical foundation for the incorporation of Dempster-Shafer theory in a forward- or backward-chaining inference system. The inclusion of probabilistic information in a resolution-based system is not yet well understood, however, and coming to grips with this problem is one of the specific goals of this project.

Theory Formation

Many problems in AI involve learning by hypothesizing, including diagnosis, planning, natural language understanding, generation of tests or experiments, and the modelling of a user, agent or environment. Programs use *bias* to select among possible inductive hypotheses or theories.

Previous AI research has formulated bias in a procedural and often *ad hoc* manner. We seek to represent the bias employed in traditional AI approaches to theory formation in a *declarative* manner, axiomatically and semantically, so as to incorporate it into the logic programming methodology. One promising approach we plan to investigate is to represent inductive theories as the result of non-monotonic reasoning, in particular circumscription [46]. We aim to apply the tools of non-monotonic reasoning to the question of when and how to weaken an overly-strong bias, once a contradiction has arisen.

We plan to investigate diagnosis, in particular diagnosis of faults in digital circuits, as an application of these theoretical ideas about theory formation. We seek to enable the use of declarative, prior knowledge beyond the design specification, e.g. the likelihood of various faults, the observables and costs of tests; as well as to provide a more principled and flexible basis for preferences among fault hypotheses, e.g. via non-monotonic reasoning and reasoning about bias, than in previous AI approaches [14, 21]

Theory Reformulation

Understanding the role of representation in problem solving has long been recognized as a central problem in AI research. The question of how to reformulate a problem description to make its solution *transparent* is at the heart of this problem. The canonical examples cited are from the world of puzzles -- the mutilated array problem and the missionaries and cannibals problem. The latter was extensively analyzed by Amarel, to identify shifts in problem representation that make the solution process more *efficient.*

We have decided to concentrate on the largely unaddressed area of problem reformulations under a given problem solving method. Within it, we seek to study the class of efficiency reformulations that can be applied to a problem specification. We will carry out this investigation in the domain of digital circuits. Given a first order logic description of a circuit at a given level of detail (which should be sufficient to solve the problem at hand), we will find a suitable reformulation of structure and behavior rules of a circuit to make a certain class of problem solving (e.g diagnosis, simulation) easier (have better space/time efficiency). This domain is chosen mainly because a preliminary analysis shows that it is amenable to the sorts of reformulations we wish to consider.

Counterfactual Implication

A type of inference that we have just recently begun to consider is that appearing in "commonsense" implication. Consider the statement, "If it hadn't been raining yesterday, we would have had a picnic." Assuming that it was in fact raining, any complete inference scheme (such as the resolution-based theorem prover in MRS) will conclude that this statement is valid. We plan to continue the formal investigation of counterfactuals already begun and will implement the results of the investigation in MRS. In light of the fact that MRS has already been used to develop diagnostic aids in the domain of digital hardware, this seems an ideal opportunity to test both the applicability and effectiveness of this use of counterfactuals. We also hope that the inclusion of a counterfactual inference mechanism in a general-purpose expert system building tool will help illuminate the precise extent of the usefulness of counterfactuals to AI generally.

SOAR: An Architecture for General Intelligence and Learning

SOAR is to be an architecture for a system capable of general intelligent behavior -- of assimilating and working on novel tasks, using diverse knowledge, learning by experience, and reflecting on its own behavior. Work to date with SOAR already provides evidence for significant advances towards attaining such an architecture. We plan to continue the development and investigation of SOAR -- to test and augment the principles on which it is built, to expand its functionality, and to have it perform a wide range of demanding tasks. Our ultimate objective is to fashion an architecture that is capable of supporting the full range of flexible activities required of intelligent behavior.

SOAR embodies a collection of mechanisms and organizational principles that express a set of distinctive hypotheses about the nature of the architecture for intelligence.

1. *Uniform task representation by problem spaces.* Every task of attaining a goal is formulated as finding a desired state in a *problem space* (a space with a set of operators that apply to a current state to yield a new state) [52]. Hence, all tasks take the form of heuristic search.

2. *Any aspect of a task as an object of goal-oriented attention.* This includes the system reflecting on its own problem-solving behavior. An exact formulation of this property requires some care, because the architecture itself is a fixed structure. The essential feature is that no domain-dependent procedures lie outside the goal system -- for implementing operators, selecting operators, analyzing situations, or anything else.

3. *Uniform representation of procedural knowledge by a production system.* SOAR is realized in a specialized production system. All satisfied productions are fired in parallel, without conflict resolution. Productions can only add data elements to working memory; the architecture is responsible for all modification and removal.

4. *Knowledge to control search is ultimately expressed in a system of preferences.* Search-control knowledge is brought to bear by the additive accumulation (via production firings) of data elements. The end-result is a set of elements called *preferences* (about the various alternatives for behaving in a problem space).

5. *All goals arise to cope with difficulties in problem solving.* Ultimately difficulties arise from a lack of knowledge about what to do next. In the immediate context of behaving, difficulties arise when problem solving cannot continue. These difficulties are detectable by the architecture, because the fixed preference decision procedure concludes successfully only when the knowledge is adequate. It fails otherwise and the architecture itself creates goals for overcoming the difficulties. This principle of operation, called *universal subgoaling*, is the most novel feature of the SOAR architecture, and many other features build upon it, e.g., automatic detection of goal attainment and learning by chunking.

6. *The basic problem-solving methods arise directly from knowledge of the task.* SOAR realizes the so-called weak methods, such as hill climbing, means-ends analysis, alpha-beta search, etc., by adding search-control productions that express, in isolation, knowledge about the task (i.e., about the problem space and the desired states). The structure of SOAR is such that there is no need for the organization of this knowledge in a separate procedural representation. This is another novel feature of SOAR.

7. *Continuous learning by experience through chunking.* SOAR learns continuously by, in effect, automatically caching all of its goal results as productions. (This mechanism appears to be directly related to the phenomenon called *chunking* in human cognition, whence its name.) It learns both operators and search control, and it produces significant transfer of learning to new situations both within the same task and between similar tasks. This ability to combine learning and problem solving has produced the most striking experimental results so far in SOAR.

Our research will have a breadth-first flavor as we seek to add major intellectual abilities to SOAR, to make SOAR robust, and to develop a theoretical foundation for the SOAR design. Only the additions to SOAR are listed below for brevity.

Chunking as a general learning mechanism

We are currently investigating two areas where chunking may be found wanting: recovering from overgeneral learning and learning from examples. The first area involves being able to learn new chunks that override previously learned chunks that were overgeneral (that is, chunks that applied inappropriately). Since SOAR only learns from experience, we are investigating ways for SOAR to retry an errorful problem-solving episode more carefully. During the retry, it may be able to override an incorrect chunk and learn new chunks that will correct that chunk in the future.

The second area involves extending chunking. While chunking is based on learning during problem solving, the inductions necessary to learn from a set of examples appear at first glance to require a quite different learning mechanism. This research effort attempts to unify learning from examples with learning while problem solving. This extension is only one of several that could be probed to test whether chunking really is a general mechanism. (Actually, the right way to pose this issue appears to be what other aspects of problem solving must be coupled with chunking to accomplish each type of learning -- where chunking operates as the final memory-modification mechanism.)

Planning

Abstraction planning appears to be a natural uniform activity in problem solving [53] and it appears to translate into a natural uniform activity in SOAR. We will concentrate our initial efforts on this type of planning, because it seems more likely to prove useful with all tasks. Initially, for tactical reasons, we will work with tasks that are already operational in SOAR, such as the *R1* configuration task. Abstraction planning, especially with the constraint of universal applicability, should provide a major challenge to SOAR, since it poses quite novel design considerations, not present initially or in the extension to chunking. If SOAR adapts gracefully to planning, we will have another major item of evidence for SOAR. Contrariwise, if major difficulties arise, we should be able to discover some important limitations to the principles on which SOAR is built.

Problem-space creation

The creation of appropriate problem spaces is a critical aspect of SOAR's performance. For SOAR, creating new and better problem spaces takes the place of creating new and better representations. So far, SOAR does not do this. The problem spaces that are used are all instances of a few general problem spaces (for resolving ties among a set of objects or for evaluating an object or operator by looking ahead in the original space) or of user-created spaces (as in the gross means-ends structure of *R1-Soar*, *Dypar-Soar*, etc.). Indeed, it came as a surprise that we were able to avoid problem-space creation as a major roadblock early in the development of *Soar1* and *Soar2*. But any substantial degree of generality for SOAR requires a powerful capability for creating problem spaces.

2.2.1.3. Resource Hardware and Core System Development

Introduction and Background

We have already explained the systematic evolution of SUMEX-AIM from its original conception as the central node for a national community of biomedical AI scientists to a more and more distributed community and computing environment. We now want to sketch our plans for the hardware and system development of the resource for the proposed new grant period.

In summary, our development efforts will build on our past experience with Lisp workstations, attempting to make a more effective and intelligent computing environment for AI research and the dissemination of AI systems out to biomedical user environments. Just as our core research and AI applications efforts are aiming for systems that will have their impact 3-5 years from now, our computing systems work aims at the hardware foundations and system facilities of the same period. Certainly the current trend toward cheaper and more powerful workstations will continue. So as these machines become more ubiquitous, we must develop the system software that will give users the tools to take advantage of these machines in all their power and flexibility. This includes the full range of tools such as text processing, electronic mail, file manipulation, budget preparation and control, drawing and so on that keep workstation users tethered to expensive and overloaded mainframe systems. But it also includes extensions so that users can interact more effectively with their computing environment through more intelligent customized interface agents and can take advantage of the networked concurrent architecture these workstations represent. We plan no changes to our mainframe hardware facilities, but will continue to operate them for the on-going work of our community as possible with decreased DRR support.

As we will be discussing more fully, the growing collection of hosts and workstations has forced AI, distributed system, and networking researchers to reexamine the question of how to use many processors on a high bandwidth local area network (LAN) most effectively. Viewed as one large interconnected system, the amount of AI research that can be done is many times more that what was possible just five years ago, but we are encountering limitations because the traditional organization of such distributed processing power in fact wastes much of this power. At present the bottleneck in the development of network-based systems has become the software, with much of the potential of the powerful workstation hardware being unrealized. The first key is to find the appropriate role for the workstations within the context of the whole network-based system [58].

Workstations and Networking

From the outset, as our research computing began moving off of mainframe computers and onto a variety of personal Lisp machines, it was clear that these systems were an integral part of a larger network environment for the development, maintenance, and distribution of software and for access to services that are only cost effective as community resources. Systems software is continually being developed by both our own staff and the Lisp machine vendors. A network system facilitates the sharing and distribution of these software efforts and servers such as large disk files, file backup systems, high quality printing, remote network gateways, and shared mainframe hosts are best shared through network interconnections.

It is not possible or desirable to run all applications on the workstation [58]. For example, large database applications require huge amounts of disk storage and some graphics or signal processing applications are processor intensive and need special hardware. Printer services require knowledge of a diverse set of fonts and special text processing languages like Impress or Postscript, and processing mail needs address resolution and domain name servers. Still further restrictions are that particular workstations are tuned to run a particular flavor of Lisp and its extensive system support environment. Consequently, workstations have been tailored for a particular processing need, and to then look for the auxiliary software and hardware requirements elsewhere. Since our research staff and users do not all reside in the same building and since Lisp machine hardware and network servers are organized around computer rooms with cable length restrictions, we cannot currently give people the needed flexibility in geographic access to use a Lisp machine from anywhere on campus or from home either.

So, when a distributed system is viewed as a collection of heterogeneous hosts comprising one interconnected system, the system as a whole has a maximum work load potential which is a function of the resources of each of the hosts in the system, and the ability of these hosts to communicate with one another via the LAN. Currently, access to such systems and effective use of their resources fall far short of the potential for at least the following reasons:

- *Lisp Machine Cost*: While costs continue to fall, the highest performance Lisp machines are still rather expensive, ranging from around $30,000 to $120,000 and this is out of the reach of many researchers. Entry into the system is through a personal workstation and we are not able to afford giving each researcher dedicated access to the best systems. In effect without flexible access facilities, the limited number of personal computers provides for rigid control on the number of users. Unlike time-sharing systems where response degrades with each added user but where there is no rigid limit to the number of users, in a distributed environment without access to a personal Lisp machine, you cannot use *any* computing resources [58]. There is currently no adequate means of sharing these workstations and consequently keeping the cost per user at a minimum, and the usage per machine at a maximum.

- *Operating System Differences*: In order to use a remote host to run a program a TELNET connection must be established with that host. The user then logs in and runs the desired programs. This implies that a user must understand the details of the executive commands and file systems of several operating systems if he wishes to take advantage of all hosts on a network to aid his research.

- *Network Protocols*: Communication between hosts on the network is by the network protocols that each vendor supplies. In our unavoidably heterogeneous computing environment, most mainframes do supply servers for some protocols but not all mainframes supply servers for all protocols. Also, some protocols may run very efficiently on a server and others may not. This is certainly the case with respect to IP/TCP versus PUP/BSP under UNIX. IP/TCP is part of the UNIX kernel and PUP/BSP runs in user space making the latter much less efficient. This inefficiency is particularly noticeable as the number of connections increases on our file servers.

- *Resource Constraints*: A user cannot easily get a picture of what the load distribution is on the combined system resources. One server or mainframe may be idle and others busy. In fact, users simply view this system as they did a time-shared mainframe. In each circumstance the researcher has important work to do, and correctly sees the underlying system as a resource to get that work done in a timely way, and often under the pressure of a deadline. Thus, they push a particular environment for all that it is worth and the limitations of these environments are exposed and often pushed to unworkable extremes. Underlying a mainframe system is an operating system and scheduler that can manage and allocate its resources as a function of the number of users. In our current system access and allocation is at best ad hoc, and for the most part managed by each user. If our timesharing experience yields any axioms, then one of would be: In any computing environment users will attempt to reach or exceed the maximum work load potential of that system. Consequently, the resources of the system must be well managed by an agent that can visualize and appropriately and effectively allocate them.

- *Remote Connection Costs*: The primary means of accessing a remote host is to establish a TELNET connection and then run jobs as if you had a direct terminal line connection to that host. Maintaining a smooth typing response over a network is very expensive and the actual processing return for the work done on both the workstation and the remote host itself per keystroke is quite small. The cost of processing one character per packet is not that much more than the cost of 512 characters per packet. The overhead is with respect to the frequency with which the packets themselves must be processed in order to give the appearance of smooth typing. Efficient management of resources should be done in such a way that typing, mouse or voice interaction, view management and screen refresh are processed on the local workstation, and that communication with the remote host is task-oriented at a high conceptual level, and, consequently, minimal.

- *Network Transparency*: The network itself is not a transparent medium of communication in the system. If a user wishes to run a job that cannot be run on his workstation, he must log onto a particular mainframe that is also connected to the network, and run his job. If he wishes to retrieve a file he must know the file server on which that file resides. The user must always be aware of the various components of the system itself. When one uses a mainframe, he need not know how many disk drives, lineprinters, CPUs, buses, or i/o channels are involved in his getting a task accomplished. It would be considered absurd for the user to have to know on which disk drive his files are stored. The mainframe hardware is transparent to the user. This should analogously apply to a networked system but in most instances does not.

- *Concurrent Process Execution*: Some tasks may take several hours or longer to complete even on the most powerful Lisp machine. There is currently no generally accessible and satisfactory way of running such a task and sharing its processing among several idle Lisp machines, even if the task is one that can be separated into distinct and independent steps. As we undertake more and more complex AI applications and as we divide tasks logically between machines (such as is proposed for the *Interviewer* and *Reasoner* parts in the dissemination of the ONCOCIN system), parallel processing and use of workstations resources becomes an essential part of the future computing environment. Projects such as the HPP Concurrent Symbolic Computing Architectures project are working on parallel system designs with orders of magnitude improvement in performance. The results of this work are a long way off, however, and in order to reach those goals, researchers require a method of more effectively utilizing concurrency in available distributed machines.

So our plan is to work on reducing these limitations, concentrating on enhancing the computing environment of Lisp workstations and more effectively exploiting their combined resources.

Central Resource Operation

Our central mainframe computers have been powerful and superb resources for the SUMEX-AIM community over the past 12 years. However, the trend toward distributed workstations is clear and it would be inconsistent for us to seek full DRR support for these central machines for another 5 years. Still, we recognize that there is a community of users, particularly young projects which need seed support prior to obtaining major funding, who will depend on the central shared mainframe for several years. Therefore, we plan a conservative and responsible phase-out of these machines. We will discontinue DRR support for the DEC 2020 demonstration machine and the shared VAX 11/780 time-sharing system starting in year 14. We will phase-out the central 2060 more slowly, budgeting 80% support for its operations in year 14 and decreasing this in 20% steps until there is no remaining DRR subsidy by year 18. This should allow ample time for remote users to find and fund alternative computing resources, most likely workstations local to their research environments.

Hardware Purchases

Our hardware purchase plans for the next grant term are modest and are aimed at maintaining access to state-of-the-art workstations for our core work. For example, Xerox has just announced a model of the 1100 series machine that is expected to sell for $18,000-19,000, run InterLisp at comparable speeds to the 1108, and have a second integrated machine able to run IBM PC software. Other machines are being designed by Texas Instruments, Hewlett-Packard, Symbolics, Japanese manufacturers, and others that will strongly influence the system goals we have for the next 5 years. Thus, we budget $75,000 per year for new workstation hardware. In the first year we will buy 4 of the new Xerox systems for use in our development efforts and as part of the ONCOCIN dissemination research. We will select future year purchases from the then available systems.

The Lisp Workstation Distributed System/Kernel

Much work has already been done on distributed computing systems that we want to take advantage of, including work in our own Stanford Distributive Systems Group [39, 37, 9]. By supporting a *distributed operating system* the workstation may perform any function best suited to the user, the hardware, and the applications at hand [58, 38, 40, 60]. An implementation of this model consists of cooperating *kernels* providing an interprocess communication system, and services implemented as processes. Related work for distributed concurrent systems has also been done using the Actor/Apiary model [32], and the Contract-Network model [67]. In the Actor/Apiary model computation is performed by independent computing elements called actors which communicate with each other by message passing. The Apiary is a networked architecture for cooperating processors. The Contract-Network model provides negotiations for not only what is to be done but also who is best suited to do it.

In our initial approach, a Lisp Workstation distributed System (LW System) will be based on the *V System* [37] but will differ in the following respects. The V system incorporates both the V kernel interprocess communication as well as a V operating system which provide a total distributed operating system for those hosts on which it runs. But each Lisp machine for which we are targeting this design already has a highly-developed operating system. Functions such as process control and memory and device management already exist on these workstations, as do the tools necessary for managing the mouse, windows, and menus. The V Kernel interprocess communication primitives, using a fixed-length synchronous message protocol, do not. In this context, processes can reside on *any* host on the LAN, and communication between any of these processes is possible. The marriage of interprocess communication with existent operating systems in this fashion provides the basis for a distributed operating system. The resulting kernel is what we will call the LW Kernel, and the resulting system the LW system.

This wedding of the V Kernel message protocol and semantics with existing and powerful Lisp machine operating systems should yield a LW system with the strengths of both systems. The LW system will be able to take advantage of the extensive work in remote process execution and virtual graphics already incorporated in hosts running the V system. For example: The V system runs on non-Lisp diskless MC-68000 based workstations that can now be purchased for $8000. We have already written applications that run in InterLisp-10 on the DEC 2060 that allow us to remotely drive the virtual graphics terminal service (VGTS) software in these diskless workstations. On a moderately loaded DEC 2060 the remote creation of views, windows, the placing of graphical objects such as text, splines, lines, and rectangles in these windows, and the interaction of menus sent from the DEC 2060 with user "mouse-buttoning" on the workstation is very responsive. By porting the remote graphics software written for the DEC 2060 to any Lisp machine and then TELNETing into that Lisp machine from a workstation either at home or on the LAN immediately allows remote access to that Lisp machine from those locations. It should be noted that *all* remote graphics is done with the interprocess message protocol, and that the amount of information necessary for all but the graphics commands involving bitmaps is minimal and therefore achievable over relatively low speed lines.

In this model, the network consists of a collection of resources accessible by *clients* and managed by *servers*. A client can be either a program or human user [37]. In this context client and server are just "roles" played by processes. For example: A user or application might make a request of a file server. Here the user/application is the client and the file server is the server. The file server then may make a request of a disk server in which case the file server becomes a client and the disk server the server.

An LW exec will run as a process on a Lisp machine, and have its own executive window for command processing. This exec will have access to the entire LW System, and thus the LW Kernel which also runs as a process. Given the above model we might have the following example: Suppose the user wishes to run SCRIBE on some server in the distributed system. The user types "SCRIBE myfile" in the LW Exec window. The LW Exec creates a client process on the local host, and this client then queries the system for the best server for running SCRIBE and blocks waiting for a reply. When a server replies the local client then opens SCRIBE as a file to execute on the remote host. If this open is successful, the server has then created the SCRIBE process which then becomes the client while the Lisp machine client becomes a server. The SCRIBE client then requests input from this server, and receives the stream "myfile" which the client opens. The client runs SCRIBE and sends the results to the server which displays them in the local window. When SCRIBE has completed it closes the transaction and goes away. The local client/server ceases to exist, and the window is left for the user to peruse, and take further action on if desired (like printing the document).

Beneath the above scenario several other transactions took place. To initiate the first client/server relationship knowledge of the server willing to run SCRIBE was necessary. To accomplish this initial rendezvous the Lisp machine client needed to first determine where to run SCRIBE, and then log onto the remote system via that server. Determining where to run a process can be done within either a *static* or *dynamic* partitioning of the underlying distributed system.

In the static partition each host has a defined set of processes it is best suited to run at initialization time, and then this is invariant over the lifetime of that configuration. Dynamic partitioning is done when load sharing over the distributed system is desired and this can often require process migration to maintain system load equilibrium. Load sharing in this sense can only be used when the systems are relatively homogenous [58]. That is to say, one cannot migrate an executable Dandelion process to a 3600 because of the inherent hardware differences, although these two systems can have a client/server relationship because the process to process communication is machine independent.

So, in our example a static partitioning means that not all systems can run SCRIBE, and only those willing servers will answer. In this simple partitioning two servers are in the same equivalence class if they provide the same services. Here we say the distributed system is partitioned with respect to *willingness*. In the dynamic partition there is one equivalence class since all hosts are essentially identical. There are other partitionings worth examining.

Consider the relationship where two servers are equivalent if they can execute the same processes. Each of the equivalence classes in this partition is then dynamically partitioned with respect to load sharing with process migration. Here for example we might have four equivalence classes: SUN 68000 workstations, Xerox D-machines, VAX's, and 3600's. Note also that the system is always partitioned with respect to willingness.

There is also a slight variation on partitioning with load sharing. In this case we first statically partition the system with respect to willingness. Then we add the following constraint: A process will be run on the *least loaded* host willing to execute that process. This simple variation makes the system responsive to overall load without process migration. Thus, in our example we would have received three replies from servers willing to open SCRIBE for execution, as well as their load averages. One can then select the system with the least load to be the server or perhaps use more intelligent planning for complex multi-step tasks, anticipating future demands. The V system currently achieves load sharing without migration by running processes on the least loaded host. In our implementation we will begin by partitioning the distributed system with respect to willingness, and then experiment with the least loaded host constraint on this partition. Ultimately we are aiming for load sharing with process migration within classes of equivalent hardware configurations. Note that concurrency can be achieved in the simplest of these schemes.

Access to the file "myfile" was also necessary. This involves locating the file, it can reside anywhere in the system, and then acquiring read access privileges. Instead of sending "myfile" the *filepath* of "myfile" would have been determined on the Lisp machine, and the SCRIBE client would have then retrieved that file from its known source. This latter server could be a file server anywhere in the LAN.

The LW Kernel has then acted as an intelligent interface between clients and servers. Beneath the kernel the roles of processes may change and this is totally transparent to the kernel itself. A kernel or server of such a distributed system acts analogously to a hardware bus, being essentially a communications switch. In addition to the physical wires used to connect modules in a hardware bus, a standard bus arbitration protocol is agreed upon to define the semantics of the communication. Analogously, in our software model, in addition to the ability to send or receive a message, a protocol is defined for the semantics of the messages [58].

Machine Independent Interprocess Message Protocol

The machine independent interprocess message protocol is used to send, receive or forward messages between processes on either the same workstation or any workstation on the LAN which implements this protocol. These messages are synchronous and in implementations like V are fixed-length to minimize overhead in both the message sender/receiver interface as well as the parser. One can for example then allocate fixed length message buffers in the kernel for message queuing. The communication between processes is intended to look like procedure calls to the sender in the sense that at the highest level a sender calls a procedure with its specified parameters, and then as a process blocks awaiting a return value in the reply message. Note that this is unlike the actor model where messaging is asynchronous. In our model a degree of synchrony can be tolerated because the frequency of messaging is very low when compared to process execution time, and if one desires concurrency a server process can be spawned and then block awaiting a reply.

In order to send a message to a process, a "token" which includes both a host identifier and process number at that host is required. At each workstation the LW Kernel supports a process registration scheme that associates a *logical process identifier* with the registrant's process identifier [37]. Processes can then query the kernel for the process identifier corresponding to a known logical process identifier. This query is supported throughout the distributed system by the means of a process-query broadcast packet. Thus, having possession of such a token is sufficient to allow the passing of a message to the associated process. On a local host the kernel's token is globally defined to enable dispatching messages to the kernel itself.

In order to implement what are essentially *call by reference* parameters, a process can pass access permission to a memory segment to the recipient of a message. This access includes read, write and execute modes as well as the address of the segment. This is primarily used for file activity and buffers associated with those files but can also be used for creating processing "locks" on critical regions and marking data areas as read or write secure in conjunction with password or special process identifier privileges.

When a message is sent by a process, ultimately that message is formulated as a token, called procedure number, and called procedure parameters in a predefined network byte order which is transparent to both the sender and recipient of the message, and then dispatched by the resident kernel. The receiving kernel will then validate the token, and queue the message in a kernel message data buffer for the receiving process. The receiving process is scheduled by the kernel and when it is called uses a kernel procedure to formulate the data in the buffer as a procedure call and simply calls that procedure if it exists. Messaging between processes can be accomplished without addressing extensive programming language issues by using fixed length interprocess messages where each field in a message also a fixed length for which 32 bits is the chosen standard. This is sufficient for both integer and pointer constants since one can implement double precision if necessary. Under some circumstances a segment of data can be appended to a message. This segment is variable up to a maximum. There is a separate data transfer facility for moving larger amounts of data [70].

Consequently, the above formulation does a syntax check within the context of the called procedures parameter specifications, ie, placing the correct number of 32 bit values on its "calling stack," and calling the procedure in that context. Such a remotely called procedure should then validate the parameters within the semantics of its properties, then execute and return a message to the caller.

For some applications it is necessary to implement the more extensive support of a chosen base language's syntax and semantics. Here programming issues such as type checking and parameter parsing must be done. The V system, for example, uses this for its remote virtual graphics terminal service (VGTS) calls. Recognizing that for interprocess communication and kernel calls a simple synchronous message exchange will do, and that for more complex applications programming language considerations must be handled is important for both efficiency and ease of implementation. Certainly, distributed kernel interaction must be simple and fast if it is to be transparent to the system as a whole, and the "process world" if you like can be defined quite easily within the file constructs that such a messaging scheme easily supports. After all, a process can be viewed as a file open for read and execution, and complicated parameters such as strings and records can be passed as a data stream when necessary. Here one simply creates a data stream pipe between two processes and allows them to send data in buffers as their applications require. Pipes can be viewed as LW System supported *standard I/O* files, and read/write requests on those files. In these latter instances type checking, if necessary, can be done in the caller/callee context thus minimizing the overhead to those contexts where it is required. Thus, the VGTS application could be structurally imposed on top of process to process pipes with the parameter passing, and type checking synchronized by the processes involved.

The LW Kernel uses this interprocess message protocol to implement those operations necessary to send, receive and forward messages between processes as well as for creating, querying, and destroying processes throughout the distributed system. This protocol is transaction oriented, each message a send/reply pair and has less load impact on client/server communications then TELNET with its continuous "sub-connection" exchanges used to maintain an open connection state. This points towards a more robust and responsive distributive system when multiple clients are running processes on the same servers.

Protocols - Uniformity Across Vendors

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; PUP/BSP, PUP/EFTP, IP/TCP, IP/TFTP, IP/UDP, and NS/SPP are those most commonly used on our LAN. PUP/BSP and IP/TCP have been used to implement both FTP and TELNET, PUP/EFTP is an Easy File Transfer Protocol on top of PUP used for boot like services, IP/TFTP is a Trivial File Transfer Protocol which uses IP/UDP datagrams, and NS/SPP is a Sequenced Packet Protocol similar to PUP/BSP and is used for FTP and TELNET. In the past we have elected to write servers for each new protocol in order to accommodate both vendor hardware and systems software. This was necessary because no one protocol has been supported on all such systems.

We are pleased that the Department of Defense IP protocol family is now supported on all hardware/operating system configurations at SUMEX and on those we anticipate purchasing in the future: IP software is available on the XEROX 1100 series workstations as of the Intermezzo system release, on Symbolics systems we have been a beta-test site for their IP software since their 5.1 operating system release, and we will be a beta-test site for the TI Explorer IP software this August. Similarly, IP is supported on all of our UNIX based file servers, and the LAN gateways route all IP datagrams.

There has been a great deal of deliberate 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 this proposed distributed operating system we are choosing to implement it on top of the IP protocol family.

In particular we are going to continue in this direction and use the IP/UDP (User Datagram Protocol). We have benchmarked all of the protocols in the above set with respect to their implementations on each of the workstations and file servers we now use. FTP using IP/TCP and PUP/BSP perform similarly on unloaded systems. They both peak at about 200K bits/sec, and this maximum is really workstation/CPU limited rather than communication bandwidth limited. On a moderately loaded UNIX based file server PUP/BSP performance begins to degrade much more rapidly than IP/TCP since the latter is implemented in the UNIX kernel and the former is not. This results in redundant copying of both the data and datagram header information from kernel to user space for the PUP/BSP code, and thus, its performance varies inversely with the system load.

The XEROX 1100 series workstations use PUP/Leaf for random file access. With Intermezzo PUP/Leaf achieves a maximum transfer rate of about 40K bits/sec on 1108's and 80K bits/sec on the 1132's. We wish to achieve transfer rates in the neighborhood of 200K bits/sec for such file access. We feel that the 1100 series are currently limited by their single priority level round-robin scheduler. Weighting all processes equally is disadvantageous in this case since the emptying of the packet input queue is handled by one of these processes, and this process is the critical path with respect to maximizing transfer rate. Using the TFTP based on IP/UDP we managed to achieve 67K bits/sec on an 1108 and 90K bits/sec on the 1132. This is quite encouraging since TFTP uses a simple packet/acknowledgment exchange for data transfer. By augmenting this algorithm to allow multiple outstanding packets we ought to achieve 100K bits/sec on the 1108's and perhaps 150K bits/sec on the 1132's within the InterLisp environment. This expectation is not overly optimistic since PUP/BSP was recently rewritten for exactly the same reason. We increased the outstanding packet window from one to four and the maximum transfer rate went from 67K to 200K in the mesa environments on these same systems. Anticipating the preemptive scheduler that XEROX is now working on,' there is no reason why the InterLisp environment cannot approach the mesa environment in these respects.

Finally, PUP's and NS packets are limited to 532 and 546 bytes of data respectively, and with IP/UDP we can essentially double this size and send packets with 1024 data bytes. This along with multiple packet windows should put the transfer rate in the neighborhood of 300K bits/sec on these systems. It is worth noting that such an IP/UDP scheme has been used between M68000 workstations on a 10-megabit net achieving a file transfer rate of 800K bits/sec. Also, the V systems downloading scheme which is encapsulated in IP/UDP datagrams achieves 400K bits/sec between a M68000 and a VAX11-780. These tests were done on lightly loaded systems.

IP/UDP is a very simple protocol with very little processing overhead. Unlike IP/TCP which allows for packet fragmentation and reassembly, IP/UDP packets are integral throughout their lifetime and ideally suited for LAN applications. Another worthwhile feature is that the simplicity of the protocol requires very little kernel management, and consequently makes multiple client/server interactions quite feasible on even a single host server without impacting either the server or distributed system loads.

The Distributed Operating System Resource Manager

The distributed operating system resource manager is an intelligent-agent that will run on a Lisp workstation with the LW Kernel. It is intended to behave in much the same way as a "pie-slice" scheduler does on a mainframe operating system except that it will have a knowledge base to govern its decisions. In its knowledge base will be a representation of the current partitioning of the distributed system and dynamic load statistics of each host in each class in the partition. Additionally, it will attempt to learn about not only each client/server type but also each process type. Different processes will impact each client/server in different ways. Understanding and dynamically adjusting to the impact processes will have on the distributed load is a difficult problem and its solution is essential in the development of the resource manager. Graphics tools for examining knowledge representations of system load with respect to clients, servers, process types and partitioning of the distributed system will be provided.

When a client wishes to run a process on the system it will query the resource manager for the best server on which to run that process rather than query-broadcast on the distributed system itself. In a simple scenario, the resource manager will select all of those servers with respect to the willingness-partition, and then select the least loaded server from this list. If a client wishes to either migrate a process from itself to a server in the dynamically partitioned system, or have a server in its class in this partition download and run a process for it, the resource manager can then mediate this transaction. It will know which servers in the class are willing to run such a process, and from this list select the server that is least loaded or better yet, maintain idle-time schedules of all such hosts and select the host that will be idle for the duration of the process execution if possible.

Certainly, centralizing the functionality of a resource manager will allow us to more clearly understand the distributed system and its interactions. Graphical representations of system, and server loads, and response to this load by the creation or destruction of processes will give us innovative insight into just what rules are necessary to manage this distributed resource. Each particular process will impact a particular server in a way that is a function of that server's hardware and operation system, and the complexity of the process and its resource requirements. Consequently, the knowledge base and rules relating its members will grow with respect to each process type as well as each server type, and as the resource manager begins to understand their interactions. Also, simply having a resource manager with a server that knows which parts of the distributed system are working at any given time will prevent a great deal of user frustration. Given the large "granularity" of processing time and the relative infrequency of communications between these processes will initially allow us to develop such a manager on an independent LISP machine. If we reach the point where the trade-off between processing time and communication load becomes critical it may be desirable to install the resource manager in several or all of the nodes in the distributed system.

Just how an intelligent-agent resource manager will behave under all instances of distributed system interaction is an excellent area for AI/distributed operating systems research.

Implementation

Initially, we plan to implement the LW Kernel on Xerox 1100 series workstations. These systems have a remarkable programming environment, and a large set of networking debugging tools to facilitate the development of the distributed kernel. We also have an excellent working relationship with the systems software group at Xerox. This will be helpful for timely acquisition of the sources for the system as well as information about any problem areas we may encounter.

The early development of the LW kernel will run in two parallel phases. The underlying IP/UDP random access file transfer protocol and the LW Kernel's interprocess message protocols (IPMP) will be done first. The former will ultimately replace the PUP/Leaf service which is a major resource drain on our UNIX file servers. This will begin to then move the 1100s towards the uniform IP network standard. Also, random file access will be an integral part of the LW System's standard I/O file access, and data transfer mechanisms. Uniformity and optimization of file transfer is important if the distributed operating system is to be responsive when servers are loaded. The LW Kernel interprocess message protocols are central and necessary for all distributed system operations. The latter and random access file I/O are initially independent mechanisms and can be developed separately.

Since the LW Kernel's IPMP are transparent with respect the the distributed system, the entire mechanism can be written and debugged on a single workstation without network interaction. The kernel runs as a process on each host in the system, and dispatches messages intended for itself and any other host in the system. All that is required to send a message to the kernel is access to the kernel's "token," and this is globally available on the workstation itself. So, initially one writes the kernel process and the primitive message dispatching stubs, *Send*, *Receive*, and *Forward*. This will be followed by process operations like *CreateProcess*, and *DestroyProcess* along with *SetProcessID*, *GetProcessID*, and *GetProcessToken*. At this time all created processes including the kernel process will be able to send/receive messages to/from each other on the workstation in exactly the same way that it would be done if these processes were distributed. Then we implement the LW System I/O protocols by beginning with the pseudo-device pipe server. A pipe is a unidirectional flow-controlled communication channel between two processes using the standard I/O protocol [37]. It is implemented via sending messages to a *pipe-server* process. This server may reside on the local host or any other host in the system so the implementation generalizes rather nicely. Each pipe is a file instance and has one reader and one writer. This may be of course the same process.

The above is written on top of the resident process scheduling and window managing functions as well as the file system. Thus calls for creating and destroying processes, opening, managing, and closing windows as well as for file system directory management already exist. The kernel process allows us to simply distribute this functionality. Once this is working on a single workstation, the software will then be loaded onto a dual system and the kernel will then use the network so that we can then run processes in a two host distributed model and debug the IPMP in this environment. Once the underlying mechanisms discussed above are working this step should be fairly easily accomplished. It reduces to insuring that the kernel's message queue can be filled via the network. The mechanisms involved are identical except that a message must be further encapsulated and then sent on the network, and the underlying network software already works.

Based on this work, it will then be appropriate to develop applications using the distributed operating system and the IP/UDP random file access protocol. The following sections discuss some of the initial applications we will explore. In later years we will work on other applications like remote file management, network performance monitoring, and more intelligent interfaces for users to systems.

Mail System

Providing an effective and responsive mail system is one of the primary goals of any modern computing environment. Most users spend at least one hour each day reading and responding to their network mail, and this now generally takes place on either the DEC 2060 mainframe or one of the UNIX systems at SUMEX. What is frustrating is that during prime computing time the routine perusal of ones mail often becomes a very time consuming task because of the load on these mainframe systems. In fact at any given moment during this time 50% of the users can be found running MM, the system mail program, on the DEC 2060. Yet, mail is a very natural function to run on an individual's workstation. To this end, it is one of the first applications directed at the LW distributed operating system.

Indeed, it makes a great deal of sense to have as much of the mail processing as possible be done on a user's workstation. This processing can be partitioned into four categories: Mail storage, Mail retrieval, Mail reading and composition, and Mail delivery. Mail storage can be done both on the local workstation and file servers. Mail retrieval involves transactions between the workstation and the storage medium. Mail reading and composition can be entirely done on the workstation, and mail delivery involves transactions between the workstation and a domain name server for address resolution, and a mail spooling service for the caching and final delivery of non-LAN mail such as that going to a site on the ARPA net and not on the LAN. Let's address each of these four areas.

Mail Storage: By mail storage we mean the storing of all *unread mail* as well as *read mail*. Initially unread mail will arrive at a file server or servers in the user's mail delivery path. This is usually accomplished by alias files on hosts that may receive mail for a person or mailing-list but on which this mail is not kept. Alias files provide a forwarding mechanism to the ultimate destination repository. In any case the mail ultimately arrives at a destination file server known to the person's resident mail process. As each letter is read it is up to the reader's discretion as to whether or not it is to remain on the workstation or be returned to the appropriate file server. Records of all mail still in the system will be kept on the file server under the user's mail account. Rereading a letter that is on the workstation can be short-circuited to remove the file server from the loop. The primary activity in this area is then the moving of mail between the user's workstation and a file server(s). This can be expedited with minimum overhead using the high transfer rate IP/UDP file service to stream the data between a client and server. Indeed, at 300K bits/sec most letters will be moved in a fraction of a second with very little impact on either the client or the server.

How this mail is arranged on the server is an important consideration if access is to be efficient and the services per letter multidimensional. On each server in the user's mail path the user will have a mail directory associated with his address at the server. The directory will be organized into a *mail spindle file, mail header file, mail keyword file* and *mail folder files*. The latter may in fact be a sub-directory on hosts supporting such a scheme.

The spindle file will have an entry for each letter. Among other things this entry will have a pointer to its header in the header file, the folder where the letter is stored, status bits indicating the state of the letter. For example: Such bits could be *seen, unseen, new, deleted, answered* and *alarm*. The alarm bit is then associated with a time-date when the user wishes to see this message's header again. Each entry has the date it was read, and the date it was answered. Finally, there will be a bit field describing key-words the owner can associate with each letter, and the associated keyword file of actual keywords. The spindle file itself will be prefixed with a header. This header will at least include time-date stamps of the last read and write access to the owner's mail, a pointer to the entry for the oldest new mail, ie, mail that has arrived since the last time the mail was read, and a pointer to the next alarm entry.

Thus, when a user first runs the mail process on his workstation the process interrogates the mail server(s) in the user's delivery path. Each such server quickly gathers the headers of the newly arrived mail, checks for any alarms that may have gone off, incorporates these headers into a message and sends them to the users workstation. The actual header file can be built in background mode as mail arrives and system resources allow to minimize this processing. Note that none of the text of the mail which is the bulk of the data has yet to be touched in this transaction.

Mail Retrieval: Mail retrieval is accomplished with a workstation client and mail/file-server server. The client is mouse driven by at least a selection process that displays active letter headers in a window. The headers which appear in this window are selected by the user with a mouse/menu interaction. When the mail client is started it probes those servers in the user's mail-path for "new" mail, ie, letters that have arrived since the last read-access to the mail spindle file. These headers will be listed in a window which has mouse interaction defined for each such header. One will be able to change the displayed headers by commands like *headers* since <date>, from <string>, to <string>, subject <string>, and all. Reading the letter associated with a header then transfers the actual text of the letter from the server to the client with a read-mail transaction, unless the letter has already been transferred to the client and is cached there. This transaction causes the read-date stamp and "seen" bit to be updated in the spindle file entry.

Mail Reading and Composition: Mail commands such as read, answer, set alarm, delete, and copy, key off of header selection. When one reads a letter it is then read from the server to the client by a read-letter transaction. The text is displayed in a window and can be scrolled as well as edited. All text editing and composition is done on the local workstation. When one answers a letter immediate destination host address recognition is mandatory. This can be accomplished by requesting host address validation after the addresses have been typed. One can use the domain name server and LAN name servers for this purpose. It also makes sense to cache known host names locally and if for some reason the name servers do not reply this list can be used for a second guess. If all else fails, then one should simply attempt to deliver the letter. If in fact the address is not valid, then this will be noted when the letter is returned to the sender as undeliverable.

Mail Delivery: Once a letter is composed and the sender requests it to be delivered, it will be spooled on one of the file/mail servers. These servers already have all of the knowledge necessary to deliver any letter to a known host. Mail delivery is done in background on these servers by a low priority process. An attempt should be made to spool the mail on the server with the smallest mail queue and such a mail-queue-size query message will be sent to those servers that respond to a request-to-send-mail broadcast. Each host can override the latter broadcast by simply remembering which servers responded to earlier broadcasts, and thus maintaining a mail-delivery-path for directing mail-queue-size queries. The system resource manager will maintain current mail delivery information. Often a host in a mail-delivery-path is down for some reason, and mailers will continuously attempt to shrink their growing mail queues by uselessly badgering this host. It makes sense to be able to request server-downtime and alternative mail routes from a resource manager. If there is no alternative route, the mail client/server can periodically check until the host comes up rather than try and send mail to a down host which amounts to useless network traffic.

Ultimately, a mail-server process ought to be able to run in the background on personal workstations, and mail could then be delivered directly to that host for those users who desire such a service. This will then take the file/mail-servers out of the mail storage and retrieval loop for such hosts. Mail is simply sent directly to the workstation that has a registered address in the domain name server tables. The mail is then retrieved and read "as if" it had already been copied from a remote file/mail-server. This latter mechanism is part of the initial design. As mail accumulates on such a host, the user will be able to take advantage of those already existent file/mail-server processes to maintain mail archive directories remotely so that old mail can still be examined in the client/server role.

Virtual Graphics Terminal Service

Virtual graphics terminal service (VGTS) allows the display of structured graphical objects on a workstation running the V system [37]. We have already indicated the power of this set of tools. While running V on a small and inexpensive workstation located either at home or on the LAN, or anywhere that has TELNET access to the LAN on which a personal Lisp machine has a TELNET server running, one can then access that Lisp machine and drive the graphics display of the smaller workstation from the Lisp machine. Geographic proximity of such a Lisp machine is then moot.

As the ratio of researchers per Lisp machine increases it is no longer possible to guarantee Lisp machine cycles to everyone during prime computing time, and a means for remotely accessing these machines in graphics mode becomes mandatory. VGTS satisfies this need perfectly. In order to install the software tools necessary for remote

VGTS access there are two requirements: First the ability to TELNET into a Lisp machine is necessary. Second, the interfacing of VGTS primitives with the current graphics/window calls on the Lisp machine. We address each of these below.

Not all of the current Lisp machines have servers which allow the establishment of an incoming TELNET connection. Currently, only the Symbolics machines have this property. What is necessary here is to modify the outgoing TELNET code where applicable so that it can also run as a server process. This is really a straightforward task. What is interesting here is just how to globally establish that the incoming data stream is to be interpreted by the Lisp machine command executive, and then all output characters are to be sent via the TELNET stream and not to the local graphics display stream. This redirection of I/O streams is well within the scope of all of our Lisp machine operating systems.

The central concept of VGTS is that application/client programs should only have to deal with creating and maintaining abstract graphical objects [37]. The actual viewing of these objects is done on the workstation running V. For example: To create a view or window on a workstation/server running V from a Lisp machine/client two things are required. The client calls a routine to remotely create a file, the *structured display file* (SDF), which will then contain descriptions of graphical objects. Each such object has an client assigned item number associated with it in the SDF. This SDF is then associated with what is commonly referred to as a window by first calling a routine to create a virtual graphics terminal(VGT) associated with this SDF, and then calling a routine to create a view on this VGT. A view is seen as a white area on the screen with a border. Thus a VGT/SDF pair can have multiple views associated with it. And one can have multiple VGT/SDF pairs at any one time as well as more than one VGT associated with the same SDF. The mapping of VGTs to SDFs need can be but not be one to one. Each of these calls involves little more than the passing of a few data bytes between the client and server.

Once the SDF/VGT relationship is established and a view is created on the server, then graphical objects can be created by adding them as items to the SDF by opening a *symbol* for editing and adding an item to that symbol in the SDF. An SDF then contains symbols which are in turns lists of items. An item itself can also be a symbol. These objects can then be displayed in the view(s) associated with the VGT. Thus, objects can appear on several VGTs at the same time. A client can also create menus on the server and then interrogate the actions implied by those menus via mouse buttoning. In fact one can actually query a mouse event within a view and receive back not only the buttons that were touched but also the VGT number and view coordinates of the cursor position itself, or a list of objects that are near the cursor position. This allows the client to interrogate, as well as edit viewed objects remotely. One need not maintain a great deal of information about objects on the client. In fact, one needs only the VGT number, SDF number, which are returned by the server at when they are created, and the item number which is sent when items are added to SDFs. A client can then inquire about this item and receive its definition as a reply. Thus, VGTS is designed to maximize what is done on the server by maintaining the SDF database and allowing detailed queries about its contents which can for the most part be driven by user/mouse interaction with their graphical representation.

The VGTS has a resident view manager for moving, zooming, opening, closing, and creating new instances of views associated with VGTs. Consequently, the view overlaying, manipulating and trimming algorithms do not impact the client. A list of the current VGTS object primitives is as follows:

Filled Rectangle These can be filled either with gray scale shades or stipple patterns or black and white monitors, and with colors on color monitors.

Horizontal Line	A simple horizontal line between two points.
Vertical Line	A simple vertical line between two points.
Point	A 2x2 pixel square.
Simple Text	A text string in a single fixed with font.
General Line	A general line between any two points.
Outline	Outlines a selected symbol with a bounding box.
Horizontal Ref	A thick horizontal line with tick marks at its end points.
Vertical Ref	A thick vertical line with tick marks at its end points.
Text	General text from varying fonts.
Raster	A general raster bit map.
Spline	A spline object which can be of interpolation order 0 to 5, open or closed, with multiple nib selection, and filled or empty.

The overhead to create each of these primitives is minimal with the exception of *Raster*. Sending large bitmaps can be expensive. Our experience has shown that user/client mouse interaction is transparent even when the "click" is sent from the server to the client and then is responded to by placing an object at the clicked position. All of this is because the bulk of the work is done on the server running V, and moving object definitions between the client and server is so efficient that the limiting factor for throughput is the CPUs involved. Thus, this is ideally suited for a Lisp machine client and a personal workstation server since neither of these is shared to any extent during the VGTS session.

The implementation of VGTS primitives on the DEC 2060 required the coding of 30 functions each averaging about ten lines of Lisp statements. An additional SDF/VGT manipulation package for maintaining a client data base, and simplifying the creation and management of SDFs, VGTs and views required about 12 pages of Lisp code. Once this was written, the writing of applications was almost trivial. Porting this part of the code to a Lisp environment is very straightforward. It has already been debugged. The real difficulty will be to interface the view and window notions of the Lisp operating system with those of the VGTS application in such a way that it is transparent to the programmer using the system. Clearly all of the graphics tools are not directly translatable to VGTS primitives. But, those that can be translated will be done in a way so that the global knowledge that the user is TELNETing to the Lisp machine will force the Lisp machine's graphics and window management software to use VGTS for creating remote windows and placing objects in those windows. In the beginning the programmer writing graphics applications to be viewed both locally and remotely will have to be aware of the constraints that the V graphics primitives impose on the nature of the objects that can be placed in a view or window. But, the development of the VGTS system has not stopped and when limitations are reached we can certainly add to the list of primitives to overcome them. This is a very promising area for exploration, and the current primitives are sufficient for most graphics applications.

Remote Process Execution

Remote process execution is inherent the the design of the distributed operating system we have been discussing. And from its initial stages the IPMP assumes the ability to transparently execute registered processes throughout this shared resource. The true area for exploration is within the dynamic partitioning of the system into classes of equivalent hardware configurations as described earlier. What is exciting is the ability to execute processes that can run on an 1100 for example by either migrating that process to another 1100 or directing that 1100 to load a particular process from a server and then run that process. This all fits nicely within the context of the IPMP. In fact, one ought to be able to cross the equivalence class boundaries and in this example have the 1100 run a compute intensive process on a 3600 to take advantage of the latter systems faster hardware. This is possible because the IPMP is defined machine independently, and the only requirements to run a process are that it be logically registered on both hosts, and the possession of its token by the 1100 client.

This is one of the most promising areas for distributed system research and can lead to true concurrency and system load sharing.

2.2.2. Collaborative Research

The details of our collaborative research projects are given in Section 6. The projects that we classify as *collaborative*, are those involving a direct interaction between our core research and the development of the specific application. These include ONCOCIN/OPAL/ONYX, MOLGEN, PathFinder, GUIDON/NEOMYCIN, PROTEAN, RADIX, and Referee. We do not include descriptions of the AIM community projects that are now using other computing resources, such as those at Rutgers, Carnegie Mellon, Pittsburgh, UC Santa Cruz, and Minnesota.

2.2.3. Service

The details of the research projects for which we provide service are also given in Section 6. The projects that we classify as *service*, are those that use the SUMEX computing resources as provided and have independent staff for developing required system components and have not been able to acquire their own computing facilities yet. These include CADUCEUS, SOLVER, CAMDA, MENTOR, RXDX, and CLIPR.

2.2.4. Training

We have an on-going commitment, within the constraints of our staff size, to provide effective user assistance, to maintain high quality documentation of the evolving software support on the SUMEX-AIM system, and to provide software help facilities such as the HELP and Bulletin Board systems. These latter aids are an effective way to assist resource users in staying informed about system and community developments and solving access problems. We plan to take an active role in encouraging the development and dissemination of community resources such as the *AI Handbook* or the *Introduction to Medical Computer Science* (see page 100), up-to-date bibliographic sources, and developing knowledge bases.

We will continue active development of the Medical Information Sciences Training and the MS:AI programs that have recently gotten underway at Stanford (see page 112). And, within our limited resources, we will accept a small number of visitors to work with our groups and learn about knowledge systems technology.

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

2.2.5. Dissemination

Our past dissemination activities speak for themselves (see page 109) and we are strongly committed to similar goals in the future. We will emphasize efforts at research software sharing and export, software commercialization, wide publication of our research results including overview analyses and retrospectives, and the presentation of selected areas of work using media like video tapes. In addition, a central part of our core research work relating to ONCOCIN is to develop more effective methodologies to disseminate AI systems into professional user communities.

2.3. Resource Organizational Structure

2.3.1. Organizational Structure

The SUMEX-AIM resource is a highly interdisciplinary research effort between the Departments of Medicine and Computer Science, as reflected in the joint Principal Investigators for the project, Professors Shortliffe and Feigenbaum. Both Professor Shortliffe and Mr. Rindfleisch, the SUMEX Director, have joint appointments between Medicine and Computer Science. The project is housed physically in the Stanford Medical Center and the principal administrative link is through the Department of Medicine. More importantly though, SUMEX is an integral part of a large and diverse AI laboratory at Stanford known as the Knowledge Systems Laboratory. The KSL comprises over 100 faculty, staff, and students working on knowledge-based systems and its overall structure, research goals, and on-going research activities are summarized in Appendix A.

2.3.2. Resource Staff Responsibilities

The resource staff is listed below with their functional roles and budgeted level of effort on the project. More details about individual roles are given in the budget justification section on page 9. This staff has long experience in developing and operating the SUMEX-AIM resource as demonstrated by our past accomplishments.

	%	ROLE IN PROJECT
RESOURCE MANAGEMENT		
E. Shortliffe	15	Principal Investigator
E. Feigenbaum	10	Co-Principal Investigator
T. Rindfleisch	70	Resource Director
L. Fagan	25	AIM Liaison and ONCOCIN Project Manager
W. Yeager	90	Assistant Resource Director
P. McCabe	75	Resource Administrator
M. Timothy	100	Secretary
Open	75	Receptionist
CORE SYSTEM DEVELOPMENT		
A. Sweer	10	Workstation development
F. Gilmurray	70	Workstation development
W. Croft	100	File service and network protocol development
R. Acuff	60	ZetaLisp/CommonLisp workstation development
C. Schmidt	60	InterLisp workstation development
N. Veizades	40	Electronics Engineer
I. Torres	40	Engineering Aid
CORE BASIC AI RESEARCH		
B. Buchanan	10	Computer Science Research Faculty
B. Hayes-Roth	15	Blackboard model control research
H. Brown	10	Concurrent blackboard architecture research
P. Nii	10	AGE retrospective
M. Hewett	40	Scientific Programmer - knowledge acquisition
P. Karp	62	Student Research Assistant
A. Garvey	62	Student Research Assistant
J. Brugge	62	Student Research Assistant
CORE ONCOCIN RESEARCH		
C. Jacobs	5	ONCOCIN Project Investigator
R. Lenon	25	Oncology Clinical Specialist
C. Lane	60	Systems Programmer - dissemination system
S. Tu	50	Scientific Programmer - EONYX development
D. Combs	50	Scientific Programmer - EOPAL and MetaOPAL
D. Vian	25	Administrative Assistant
J. Rohn	100	Data Manager
A. Grant	50	Secretary
T. Barsalou	62	Student Research Assistant
L. Perreault	62	Student Research Assistant
SYSTEM OPERATIONS SUPPORT		
R. Tucker	20	Operations Manager
P. Ryalls	20	System Manager and User Support

2.3.3. Resource Operating Procedure

The mission of SUMEX-AIM, locally and nationally, entails both the recruitment of appropriate research projects interested in medical AI applications and the catalysis of interactions among these groups and the broader medical community. These user projects are separately-funded and autonomous in their management. They are selected for access to SUMEX on the basis of their computer and biomedical scientific merits, as well as their commitment to the community goals of SUMEX. Currently active projects span a broad range of applications areas such as clinical diagnostic consultation, molecular biochemistry, molecular genetics, medical decision making, and instrument data interpretation (see section 6).

2.3.3.1. New Project Recruiting

We continue our active search for new AI applications to biomedicine and, within the limits of our machine and manpower resources, are recruiting pilot projects to replace projects that have matured and moved off of the SUMEX-AIM machine. Information about SUMEX-AIM is available through well-attended presentations at national conferences in Artificial Intelligence, such as AAAI-M, and interest in the AI approach to medical decision making has strongly increased in the national medical computing conferences. SUMEX-AIM related researchers are often the key personnel at these presentations. Our dissemination efforts and the AIM workshops also provide broad exposure to our work in recruiting new and interesting projects.

The criteria for the acceptance of new pilot projects continues to concentrate on the potential for excellence, and the novelty of the proposed concepts. We continue to seek projects that will extend our understanding of basic science issues underlying the application of the artificial intelligence approach to medical decision making. Thus, a project that will break new ground will be preferred to a project that uses existing ideas in a new area of medicine. We also encourage pilot projects to collaborate with the existing bases of expertise in artificial intelligence techniques. Developing a new pilot project now requires more background and understanding of previous work in AI in medicine. However, the time needed to build a first prototype version may be substantially decreased by the use of packages developed by other SUMEX-AIM projects. SUMEX-AIM provides a unique opportunity for the development of pilot projects. We hope to build the number of pilot projects consistent with SUMEX resources and the availability of worthy project proposals.

2.3.3.2. Stanford Community Building

The Stanford community has grown significantly and we have undertaken several internal efforts to encourage interactions and sharing between the projects centered here. The positions of Professor Shortliffe and other collaborators in the School of Medicine provides frequent exposures of SUMEX-AIM work to medical colleagues to stimulate thinking about new application areas. Weekly informal lunch meetings (SIGLUNCH) also are held between community members to discuss general AI topics, concerns and progress of individual projects, or system problems as appropriate. In addition, presentations are invited from a substantial number of outside speakers. Finally, the MIS and MS:AI special degree programs supply a continuing flow of good new students to work on novel applications.

2.3.3.3. Existing Project Reviews

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

2.3.3.4. Resource Allocation Policies

Policies have been established to control the allocation of critical facility resources (file space and central processor time) on the SUMEX-AIM 2060. File space management begins with an allocation of file storage, defined for each authorized project in consultation with the management committees. This allocation for any given project is redistributed among project members as directed by the individual principal investigators. System enforcement of project allocations is done on a weekly basis. We are using the TOPS-20 class scheduler provide an *a priori* 40:40:20 allocation of CPU time among national projects, Stanford projects, and system development. In practice, the 40:40 split between Stanford and non-Stanford projects is only approximately realized (see Figure 15 on page 296 and the tables of recent project usage on page 298).

Our job-scheduling controls bias the allocation of CPU time according to the 40:40:20 community split but the controls are "soft" in that they do not waste computer cycles if users below their allocated percentages are not on the system to consume those cycles. In the early years, the operating disparity in CPU use reflected a substantial difference in demand between the Stanford community and the developing national projects, rather than inequity of access. This disparity in usage disappeared in recent years with the growth of the national user community. Now, because of the availability of significant additional computing resources at other AIM sites and the growing demand of the Stanford community the allocation gap is widening again. We will continue to exercise the nominal 40:40:20 controls to facilitate national access to the machine.

Our system also categorizes users in terms of access privileges. These include fully-authorized users, pilot projects, associates, guests, and network visitors in descending order of system capabilities. We want to encourage bona fide medical and health research people to experiment with the various programs available with a minimum of red tape, while not allowing unauthenticated users to bypass the advisory group screening procedures by coming on as guests. So far, we have had relatively little abuse compared to that experienced by other network sites, perhaps because of the personal attention directed by senior staff to logon records, and to other security measures. However, experience behooves us to be cautious about being as wide open as might be preferred for informal service to pilot efforts and demonstrations.

2.3.4. Support of Service and Collaborative Projects

We have pondered the possibilities of a fee-for-service approach for allocation of the resource in the coming period. We believe that this would be inappropriate for an experimental research resource of national scope like SUMEX for several reasons:

1. We have based the development of the national SUMEX-AIM resource entirely on experimentation with tools for new AI research and inter-community scientific collaborations. If obliged to recover some portion of the overall facility cost, these goals may become diluted with administrative and financial impediments, and commitments to paying users, that would be tangential to our main research efforts. There is little doubt that a facility of the quality of SUMEX could be tailored to attract paying users (we have turned down numerous such potential users already because they were not aligned with our AI research goals). However, there is little point in demonstrating once again that a computing resource can pay for itself. Rather we should judiciously allocate the available resources to encouraging new medical AI research efforts and stimulating scientific collaborations that cannot always be financially justified at these early stages.

2. A key element in our management plan for SUMEX is to encourage mature projects to acquire computing resources of their own, as soon as justified, and to couple them through communications tethers to SUMEX. This preserves the limited capacity of the central resource for new research efforts and applications. Maturing projects (those able to pay a fee) have every incentive to obtain separate facilities since they cannot obtain sufficient resources from the heavily loaded central resource. In this way such projects effectively pay a "fee" in securing their own facilities and freeing up part of the central facility.

3. A fee structure would impose substantial additional administrative overhead on the project, compounded by its national character. We would face problems of accountability for the transfer of funds from one institution to another. Also SUMEX is a evolving research resource based on changing experimental facilities. Any fee schedule would need to change frequently to fairly respond to developments in the system. Put simply, it would be an administrative nightmare.

For these reasons, we plan to continue indefinitely our present policy of non-monetary allocation control. We recognize, of course, that this accentuates our responsibility for the careful selection of projects with high scientific and community merit.

2.3.5. Resource Advisory Committee

Since the SUMEX-AIM project is a multilateral undertaking by its very nature, several management committees have been created to assist in administering the various portions of the SUMEX resource. As defined in the SUMEX-AIM management plan adopted at the outset in 1974, the available facility capacity is allocated 40% to Stanford Medical School projects, 40% to national projects, and 20% to common system development and related functions. Within the Stanford aliquot, Profs. Shortliffe and Feigenbaum have established an advisory committee to assist in selecting and allocating resources among projects appropriate to the SUMEX mission. The current membership of this committee is listed in Appendix C.

For the national community, two committees serve complementary functions. An *Executive Committee* oversees the operations of the resource as related to national users and renders final decisions on authorizing admission for new projects and revalidating continued access for existing projects. It also establishes policies for resource allocation and approves plans for resource development and augmentation within the national portion of SUMEX (e.g., hardware upgrades, significant new development projects, etc.). The Executive Committee oversees the planning and implementation of the AIM Workshop series, and assures coordination with other AIM activities as well. The Committee will continue to play a key role in assessing the possible need for additional future AIM community computing resources and in deciding the optimal placement and management of such facilities. The current membership of the Executive Committee is listed in Appendix C.

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

These committees have actively functioned in support of the resource. Except for meetings held during the AIM workshops, the committees have "met" by messages, net-mail, and telephone conference, owing to the size of the groups and to save the time and expense of personal travel to meet face-to-face. The telephone meetings, in conjunction with terminal access to related text materials, have served quite well in accomplishing the agenda business. Other solicitations of advice requiring review of sizeable written proposals are done by mail.

We will continue to work with the management committees to recruit the additional high-quality projects which can be accommodated and to evolve resource allocation policies which appropriately reflect assigned priorities and project needs. We will continue to make information available about the various projects both inside and outside of the community and thereby promote the kinds of exchanges exemplified earlier and made possible by network facilities.

3. Impact of Current Biomedical Problems

We have already discussed the importance and impact of the work of the SUMEX-AIM community in our section about "Significance" (see page 69). In summary, the impact of our work is as widespread as the applications being pursued. Besides the intrinsic intellectual importance to computer science, SUMEX-AIM has had and will continue to have a strong effect on clinical diagnostic aids (e.g., MYCIN, CADUCEUS, and CASNET), on clinical decision making (e.g., ONCOCIN, MDX, SOLVER, and ATTENDING), on biochemistry (e.g., DENDRAL, SECS, CRYSALIS, and PROTEAN), on molecular biology (e.g., MOLGEN and BIONET), on cognitive psychology (e.g., ACT, CLIPR, SCP, and SOAR), on the training of health care and computer science professionals (e.g., through the MIS, PhD, and MS:AI programs), on the development of an active national community of research work in this area, and on the rapid growth of commercial of AI systems based to a significant degree on SUMEX-AIM work (e.g., DENDRAL, EMYCIN, UNITS, SECS, and MAINSAIL).

4. Institutional Development

On the research side, the SUMEX-AIM resource has been the key element in the development of the entire knowledge engineering program at Stanford. Starting with a handful of people in 1974, the KSL now numbers over 100 active research faculty, staff, and students (see page 285). The broad array of projects we have undertaken entail significant interdisciplinary collaborations made possible by SUMEX. The critical mass of this work is fueling still more growth, limited by manpower and physical facilities.

On the instructional side, SUMEX-AIM has both encouraged and made possible the development of special degree programs such as the MIS and MS:AI programs (see page 112), in addition to the active computer science PhD program at Stanford.

5. Future Plans

The SUMEX-AIM resource has been in existence for almost 12 years and while significant progress has been made in the study of artificial intelligence and its applications to biomedicine, much remains to be done (see page 118). AI is among the most difficult research areas in its own right and the effective penetration of AI technology into biomedicine is difficult as well because of the scale of health care problems, the knowledge intensiveness of most application areas, and the management and professional issues surrounding patient responsibility in health care delivery. All of these factors mean that research in biomedical AI will be a long term program and resources such as SUMEX-AIM will continue to play an essential role in facilitating this work, even beyond the 5 year plan of this proposal.

6. Collaborative and User Projects

The following sections report on the community of collaborative and user projects and "pilot" efforts, including local and national users of the SUMEX-AIM facility at Stanford. However, those projects admitted to the National AIM community and using other computational resources for their work are not explicitly reported here (see page 116).

In addition to these detailed progress reports, abstracts for fully-authorized projects can be found in Appendix D on page 311.

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

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

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

III. RESEARCH PLANS
   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

In addition this year, we asked a more specific set of questions regarding the role and need for a centralized SUMEX-AIM resource that has guided our renewal plans:

- What do you think the role of the SUMEX-AIM resource should be for the period after 7/86, e.g., continue like it is, discontinue support of the central machine, act as a communications crossroads, develop software for user community workstations, etc.

- Will you require continued access to the SUMEX-AIM 2060 and if so, for how long?

- What would be the effect of imposing fees for using SUMEX resources (computing and communications) if NIH were to require this?

- Do you have plans to move your work to another machine or workstation and if so, when and to what kind of system?

Collaborative and User Projects

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.

6.1. Stanford Projects

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

6.1.1. 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. This work derived from our first system, the MYCIN program. That research led to three sub-projects (EMYCIN, GUIDON, and ONCOCIN) described in previous annual reports. EMYCIN has been completed and its resources reallocated to other projects. GUIDON and ONCOCIN have become projects in their own right.

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, will become 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 are using NEOMYCIN as a test bed for the explanation capabilities that will eventually be part of our instructional programs.

The purpose of the current contracts is to construct an intelligent 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 will have 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 utilized in additional ways that are medically relevant. Using the knowledge to teach medical students is perhaps foremost among these, and NEOMYCIN continues to focus 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 will in the next year begin to collaborate more closely with the medical community to design an effective, useful tutoring program. Stanford Medical School faculty, such as Dr. Maffly, have shown considerable interest in this project. A research fellow associated with Maffly, Curt Kapsner, M.D., joined the project two years ago to serve as medical expert and liaison with medical students at Stanford.

C. Highlights of Research Progress

C.1 Accomplishments This Past Year

C.1.1 The NEOMYCIN Consultation Program

NEOMYCIN is distinguished from other AI consultation programs by its use of an explicit set of domain-independent metarules for controlling all reasoning. These rules constitute the diagnostic procedure that we want to teach to students: the stages of diagnosis, how to focus on new hypotheses, and how to evaluate hypotheses. This diagnostic procedure as well as the knowledge base underlying the procedure has remained relatively stable this year. Our work in explanation highlighted the importance of making the knowledge used by the system at all levels as explicit as possible. As a result, this year we have extended and refined a previous predicate calculus representation of NEOMYCIN's metaleval rules. To avoid earlier problems of efficiency with this representation, we have also written a compiler that produces Lisp code from our predicate calculus notation. As a result, we are able to run the more efficient Lisp code and use the explicit notation for explanation and modeling.

To develop and test our model of heuristic classification, we are producing from NEOMYCIN a generic system, called HERACLES, that can be used to solve other problems by classification. This is an "E-NEOMYCIN," NEOMYCIN without its current medical knowledge. HERACLES is a variant of EMYCIN; it enables a knowledge engineer to produce NEOMYCIN-like knowledge bases containing the NEOMYCIN diagnostic procedure and domain knowledge organization. To prove its true generality, our first HERACLES knowledge base is in the manufacturing domain, for diagnosing sand casting problems (for the process of forming metal objects using sand molds). Future knowledge bases could be drawn from many medical and non-medical domains.

C.1.2 The ODYSSEUS Modeling System

This effort concerns automation of the transfer of expertise between an expert system and a human expert. A major goal is to produce a system that can watch an expert solve a problem and automatically recognize *differences* between the expert's underlying knowledge base and an expert system's knowledge base. This system should demonstrate how a knowledge of these differences can aid knowledge acquisition and intelligent tutoring. The program implementing this approach, called ODYSSEUS, has several stages of operation. Based on a large set of problem-solving sessions, the program first induces the rule and frame knowledge to drive HERACLES. Using this initial knowledge base as a "half-order theory," subsequent problem-solving sessions are tracked step by step: for each observable step the specialist makes, ODYSSEUS generates and scores the alternative *lines of reasoning* that can explain the specialist's reasoning step. When no plausible reasoning path is found, or all found ones have a low score, the program assumes it is deficient in either its strategic or domain knowledge. It attempts to acquire the missing knowledge either automatically or by asking the specialist specific questions. In a variation, the specialist justifies each problem-solving step using the vocabulary of an abstract justification language. These justifications aid in scoring alternative plausible lines of reasoning.

Each of the stages of ODYSSEUS has been implemented as a separate subsystem. These subsystems are now being integrated.

C.1.3 The NEOMYCIN Explanation System

The initial explanation system of NEOMYCIN enables the user to ask WHY and HOW questions during a consultation. That is, when the program prompts the user for new data, the user may ask WHY the data is being requested or HOW some strategic task will be (or was) accomplished. Unlike MYCIN's explanation system, upon which this kind of capability is patterned, explanations in NEOMYCIN are in terms of the diagnostic plan, not just specific associations between data and diagnoses.

The next phase of this work is to answer WHY questions by condensing the entire line of reasoning. The program uses general explanation heuristics, models of the user's knowledge of diseases and of strategy, and a history of the user's interaction with the current consultation to select the task, focus, and domain information that is most likely to be of interest. Some of the heuristics used by the explanation system include: 1) mentioning the last task whose focus (or argument) changed in kind (e.g. from a disease hypothesis to a finding request); 2) never mentioning tasks that are merely iterating over a list of rules, findings, or hypotheses; and 3) only mentioning tasks with rules as an argument to programmers. These heuristics, as well as the general procedure for providing explanations, have been implemented in the same task and metarule language used to represent NEOMYCIN's diagnostic strategy. In addition, the explanation system has been extended to use the MRS version of the task metarules. We are thus able to select the specific medical relations that were used by the metarule in determining what action to take. As a result, we have more detailed and concise information to explain to the user. The clearer representation of both the information that can be explained and the explanation procedure provides us with a flexible, explicit encoding of our method for producing explanations, which will serve as a basis for devising tutoring techniques, as well as understanding explanations provided by users of their diagnostic strategy.

Related to our explanation condensation is an effort to teach the strategic language of tasks to students. For example, we will have students annotate a NEOMYCIN transcript in terms of tasks and foci, to help them recognize good strategic behavior. This requires a common language of what the tasks are, e.g. "grouping" and "asking general questions." Rather than just marking annotated tasks, we seek the *principles* by which the tasks could be consistently structured into primitives and auxiliary. These same principles could be used by the explanation system for choosing tasks to mention. Our current theory is that these primitive, or "interesting," operations correspond to metarules that establish a new focus.

C.1.4 Graphics for Teaching

We are continuing to make extensive use of graphics in our programs. As part of our series of instructional programs, GUIDON-WATCH has been implemented as a graphics system for watching NEOMYCIN's reasoning. For example, we can highlight the hypothesis under consideration in the diagnostic taxonomy and show graphically how the program "looks up" its hierarchies before refining hypotheses. In addition, the user is able to explore the findings, hypotheses, rules and tasks that comprise the knowledge base, see selected causal association networks, view the differential as it changes, and keep track of hypotheses with evidence and positive findings. All of these can be easily selected with a consistent menu system, and windows on the screen are automatically organized to clearly display the information requested by the user.

C.2 Research in Progress

The following projects are active as of June 1984 (see also near-term plans listed in Section III.A):

1. development of a prototype of a bottom-up student modeler
2. standardization of display code
3. prototype of GUIDON-MANAGE
4. prototype of HERACLES and demonstration in non-medical domain
5. user model incorporated in explanations, with summarization
6. student model learning discrepant domain knowledge

D. Publications Since January 1984

1. Clancey, W. J.: *Knowledge acquisition for classification expert systems.* Proc. ACM-84. Also Heuristic Programming Project Report HPP 84-18, Computer Science Dept., Stanford Univ., July, 1984.
2. Clancey, W.J.:*Heuristic classification.* Knowledge Systems Laboratory Report KSL 85-5, Computer Science Dept., Stanford University, March 1985.
3. Richer, M., and Clancey, W.J.: *GUIDON-WATCH: A graphic interface for browsing and viewing a knowledge-based system.* Submitted to IEEE.
4. Wilkins, D.C., Buchanan, B.G., and Clancey, W.J.: *Inferring an expert's reasoning by watching*. Proc. 1984 Conference on Intelligent Systems and Machines, Rochester, MI, April 1984, pp.51-58.

E. Funding Support

Contract Title: "Exploration of Tutoring and Problem-Solving Strategies"
Principal Investigator: Bruce G. Buchanan, Prof. Computer Science, Research
Associate Investigator: William J. Clancey, Research Assoc. Computer Science
Agency: Office of Naval Research and
Army Research Institute (joint)
ID number: N00014-79-C-0302
Term: March 1979 to March 1985 (renewal proposal pending)
Total award: $683,892

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Medical Collaborations and Program Dissemination via SUMEX

A great deal of interest in GUIDON and NEOMYCIN has been shown by the medical and computer science communities. We are frequently asked to demonstrate these programs to Stanford visitors or at meetings in this country or abroad. GUIDON is available on the SUMEX 2020. Physicians have generally been enthusiastic about the potential of these programs and what they reveal about current approaches to computer-based medical decision making.

B. Sharing and Interaction with Other SUMEX-AIM Projects

We plan to add learning capabilities of two forms into this framework, involving interactions with the machine learning group within the KSL and Prof. Paul Rosenbloom's project on SOAR.

GUIDON/NEOMYCIN retains strong contact with the ONCOCIN project, as both are siblings of the MYCIN parent. These projects regularly share programming expertise and continue to jointly maintain large utility modules developed for MYCIN. 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 staff has been extremely helpful in maintaining connections between Xerox D-machines and SUMEX. The SUMEX staff also rewrote communication software used to link the D-machines to SAFE, the file saver used by the GUIDON/NEOMYCIN group. This has greatly improved both performance and reliability.

III. RESEARCH PLANS

A. Project Goals and Plans

Research over the next year will continue on several fronts, leading to several prototype instructional programs by early 1986.

1. Test student modeling program on cases chosen for teaching, collecting data for further development of the program, as well as exploring the range of student approaches to diagnosis.

2. Extend the explanation system to do full summaries. Incorporate modeling capabilities that relate inquiries to a user model. Provide explanations tailored to this interpretation of the motivation behind the user's inquiry.

3. Extend student modeling system to include heuristics for generating tests that will confirm and extend the model. Improve the model to include analysis of patterns in model interpretations, including dependency-directed "backtracking" in the belief system and some capability to critique the modeling rules. Relate this to knowledge acquisition research.

4. Work closely with medical students to package NEOMYCIN capabilities in a "workstation" for learning medical diagnosis, determining what mix of student and program initiative is desirable.

5. Refine NEOMYCIN diagnostic model (relations and procedures) by student modeling and knowledge acquisition efforts.

6. Develop, debug, and document an exportable version of HERACLES, a generic knowledge engineering tool that can be used to produce additional medical and non-medical knowledge bases to be tutored by GUIDON2.

7. Formalize heuristics for teaching, given the NEOMYCIN model and heuristics for explanation and modeling, embodied in different versions of GUIDON2.

B. Long term plans: the GUIDON2 Family of Instructional Programs

We sketch here our general conception of the research we plan for 1985-88, specifically the GUIDON2 family of instructional programs, based on the NEOMYCIN problem-solving model. Our ideas are strongly based on recent proposals by J.S. Brown, particularly his paper "Process versus Product -- A perspective on tools for communal and informal electronic learning" and some related papers that he wrote in 1983, in which he proposes methods for giving a student the ability to reflect on how he solved a problem. We have designed a family of seven programs that as a sequence will teach students to think about their own thinking process and to adopt efficient, effective approaches to medical diagnosis.

The key idea is that NEOMYCIN provides a *language* by which a program can converse with a student about strategies and knowledge organization for diagnosis. NEOMYCIN's tasks and structural terms provide the *vocabulary* or *parts of speech*; the meta-rules are the *grammar* of the diagnostic process. We will construct different graphic, reactive environments in which the student can observe, describe, compare, and improve his own diagnostic behavior and that of others. By "reactive environment" we mean that these programs are not passive, they will watch what the student does, build a model of his understanding and learning preferences, and provide corrective advice.

Our approach is to delineate different kinds of interactions that a student might have with a program concerning diagnostic strategies. Thus, each instructional system has a name of the form GUIDON-<student activity>, where the name specifies what the student is doing (e.g., watching, telling). The programs can be made arbitrarily complex by integrating coaches, student models, and explanation systems. There are many shared, underlying capabilities that will be constructed in parallel and improved over time. We try here to separate out these capabilities, trying to get at the minimum interesting activities we might provide for a student.

GUIDON-WATCH The simplest system allows a student to watch NEOMYCIN solve a problem, perhaps one supplied by the student. Graphics display the evolving search space, that is, how tasks, as operators, affect the differential (Differential ---(Question X)---> Differential'). The student can step through slowly and replay the interaction. He can ask for prose explanations and summaries of what the program is doing. The program will also indicate its task and focus for each data request. This introduces the student to the idea that the diagnostic process has structure and follows a certain kind of logic. The graphic capabilities of this program are nearly complete.

GUIDON-MANAGE In this system the student solves a problem by telling NEOMYCIN what task to do at each step. Essentially, the student provides the strategy and the program supplies the tactics (meta-rules) and domain knowledge to carry out the strategy. The program will in general carry through tasks in a logical way, for example, proceeding to test a hypothesis completely, and not "breaking" on low-level tasks that mainly test domain knowledge rather than strategy. The program will not pursue new hypotheses automatically. However, the student will always see what questions a task caused the program to request, as well as how the differential changes. This activity leads the student to observe what a strategy entails, helping him become a better observer of his own behavior. Here he shows that he knows the structural vocabulary that makes a strategy appropriate.

GUIDON-ANNOTATE This system allows the student to annotate a NEOMYCIN typescript, *explaining* in strategic and/or domain terms what the program is doing each time it requests new case data, indicating the task and focus associated with each data request. The program will indicate, upon request, where the student is incorrect and which annotations are different from NEOMYCIN's, but are still reasonable interpretations. The student will be able to choose these tasks from a menu of icons, either linearly or hierarchically displayed, as he prefers. (Again, NEOMYCIN will annotate its own solutions upon request and allow replaying.) This activity gets the student to think strategically by recognizing a good strategy. In this way, he learns to recognize how strategies affect the problem space.

GUIDON-APPRENTICE This is a variant of NEOMYCIN in which the program stops during a consultation and asks the student to propose the next data request(s). The student is asked to indicate the task and focus he has in mind, plus the differential he is operating upon. The program compares this proposal to what NEOMYCIN would do. In this activity we descend to the domain level and require the student to instantiate a strategy appropriately. Ultimately, such a program will use a *learning model* that anticipates what the student is ready to learn next and how he should be challenged. Early versions can simply use built-in breakpoints supplied by an expert teacher. In the future, programs will develop their own curriculums from a case library.

GUIDON-DEBUG Here the student is presented with a buggy version of NEOMYCIN and must debug it. He goes through the steps of annotating the buggy consultation session, indicating what questions are out of order or unnecessary, indicating what tasks are not being invoked properly, and then trying out his hypothesis on a "repaired" system. He is asked to predict what will be different, then allowed to observe what happens. This activity teaches the student to recognize how a diagnostic solution can be non-optimal, further emphasizing the value of good strategy. It also provides him with key meta-cognitive practice for criticizing and debugging problem behavior. With time, GUIDON will collect examples of buggy student behavior, providing a library of pitfalls to be shown to new students.

GUIDON-SOLVE This is the complete tutorial system. The student carries through diagnosis completely, while a student modeling program attempts to track what he is doing and a coach interrupts to offer advice. Here annotation, comparison, debugging, and explanation are all integrated to illustrate to the student how his solution is non-optimal. For example, the student might be asked to annotate his solution after he is done; this will point out strategic gaps in his awareness and provide a basis for critique and improvement. A "curriculum" based on frequent student faults and important things to learn will drive the interaction. In this activity, the student is on his own. Faced with the proverbial "blank screen," he must exercise his diagnostic procedure from start to finish.

GUIDON-GAME Two or more students play this together on a single machine. They are given a case to solve together, and each student requests data in turn. All students receive the requested information. When a student is ready, he makes a diagnosis, indicated secretly to the program while the others are not watching. He then drops out of the questioning sequence. However, he can re-enter later, but of course will be penalized. Afterwards, score is based on the number of questions asked and use of good strategy. The coach will indicate to weak players what they could learn from strong players, encouraging them to discuss certain issues among themselves. *Variation*: one person solves while one or more competing students annotate the solution and show where it could be improved. *Variation*: one team introduces a bug into NEOMYCIN (and predicts the effect), and the other team finds it (as in SOPHIE). This activity will encourage students to share their experiences and talk to and learn from each other.

C. Requirements for Continued SUMEX Use

Although most of the GUIDON and NEOMYCIN work is shifting to Xerox Dolphins and Dandelions (D-machines), the DEC 2060 and 2020 continue to be key elements in our research plan. Our primary use of the 2060 will be to develop the NEOMYCIN consultation system, possibly by remote ARPANET access. Because of address space limitations, the consultation program can be combined with explanation or student modeling facilities, but not both, as is required for GUIDON2 programs. We continue to use the 2020 for demonstrating the original GUIDON program. As always, the 2060 will be essential for work at home, writing, and electronic mail.

D. Requirements for Additional Computing Resources

With the addition of two new D-machines for this work, our computing needs will be adequately met in the coming 1-2 years at least.

The D-machine's large address space permits development of the large programs that complex computer-aided instruction requires. Graphics enable us to develop new methods for presenting material to naive users. We also plan to use the D-machine as a reliable, constant "load-average" machine, for running experiments with physicians and students. The development of GUIDON2 on the D-machine will demonstrate the feasibility of running intelligent consultation or tutoring systems on small, affordable machines in physicians' offices, schools, and other remote sites.

E. Recommendations for Future Community and Resource Development

As we shift our development of systems to personal Lisp machines, such as the Dolphin, it becomes more difficult to access these programs remotely for access from our homes (so that we may work conveniently during the evenings and weekends) and from remote sites for collaboration and demonstration. This problem will be partly ameliorated by "dial-up" (modem) access to these machines, but the use of bitmapped displays requiring a high bandwidth makes the phone lines inadequate for our purposes. Further technological development of networks, probably involving access over cables, will be necessary.

As computer resources become more distributed, the need for a central machine does not diminish. Programs and knowledge bases continue to be shared, requiring high-speed network connections among computers and file servers. SUMEX-AIM's role will shift slightly over the next few years to accommodate these needs, but its identity as a central resource will only change in kind, not importance. Moreover, sophisticated printing devices, such as the Xerox RAVEN, must necessarily be shared, again using a network. Maintenance of this network and its shared devices will become a key activity for the SUMEX staff. Thus, while computing resources will be provided by the "outboard engines" of personal machines, the community will remain intricately linked and dependent on common, but peripheral, resources.

From this perspective, future resource development should focus on improving the capabilities of networks, file servers, and attached devices to respond to individual requests. Multi-processing becomes a necessity in such an environment, so a request can be honored while the user returns to continue his programming or editing.

6.1.2. MOLGEN Project

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

Prof. E. Feigenbaum and Dr. P. Friedland
Department of Computer Science
Stanford University

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 200 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 400 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 1000 academic molecular biologists in the United States.

C. Highlights of Research Progress

C.1 Accomplishments

The current year has seen the completion of our initial study of the Yanofsky project on genetic regulation in the *trp* operon. In addition we have tested several models of qualitative simulation of biological systems and begun our design of a theory discovery system. Finally, a new application program for DNA sequence analysis was developed by one of our research collaborators. The highlights of this work are summarized in several categories below.

C.1.1 The Scientific Process of Theory Formation, Modification, and Testing

The first goal of our work in scientific theory discovery was to extensively study an existing example of the process. Professor Charles Yanofsky's work in elucidating the structure and function of regulation in the trp operon of E. coli provided us with an excellent subject that spanned twelve years of research, dozens of collaborators, and almost one hundred research papers.

We have conducted extensive interviews with Professor Yanofsky and many of his former students and collaborators. We have examined most of the relevant research papers. We believe we now have a good understanding of the three major classes of knowledge that were important in the discovery of the theory of regulation in the trp operon: knowledge about the relevant biological objects, knowledge about the techniques used to elicit new information, and discovery heuristics used to build new models.

In addition, we have developed an initial model for the inference mechanisms used during the discovery process. This model includes at least four different types of reasoning: data-driven, theory-driven, analogy to closely-related biological systems, and analogy to other systems (railroad engines and tracks, for example).

C.1.2 Knowledge-Based Simulation of the Trp Operon

The first major programming task of our project was to build a knowledge base representing the initial state of knowledge about the tryptophan operon system at the beginning of the Yanofsky research. This initial knowledge base contains information relevant to genetic regulation in general and to the trp operon system in particular. The information relates both to structure, i.e. the physical characteristics of the biological objects, and to function, i.e. the operational characteristics of the biological objects. In addition, the procedural knowledge needed to relate structure to function plays an important part in the knowledge base.

The goal was to have a knowledge base that can be used "actively" to simulate the result of various possible changes in the underlying regulatory model. For example, a common experimental method for studying a biological system is to introduce a mutation which destroys the functionality of some piece of the system. The regulatory knowledge base should be able to simulate and describe the results of such a "deletion mutation."

As a first experiment, we built the knowledge base using the Unit System (developed under previous MOLGEN work). We were able to successfully model most of the important processes of Jacob-Monod repression, the initial model of genetic regulation used in the Yanofsky research.

C.1.3 A Model for Theory Discovery

In parallel with our work on knowledge base construction, we designed an initial architecture for theory proposal, extension, and correction. In human scientists we have observed at least four major types of reasoning during the cognitive process. The first is data-driven reasoning when the major goal is to explain individual experimental results. The second is theory-driven reasoning which occurs when a partial theory or model drives its own extension. The third type of reasoning involves looking at closely related biological systems (e.g, noticing a similar behavior in the his operon system). The final type of reasoning relates to more distant analogies; thinking of DNA polymerase moving along a nucleotide sequence as similar to a railroad engine moving along a set of tracks. Our discovery system architecture embraces all of these reasoning types within a blackboard-style hybrid architecture.

In addition, we have fit our overall model of simulation and discovery into a framework of research on machine learning. This framework involves interacting performance and learning elements. The performance element, here the knowledge-based system for qualitative simulation of regulatory genetics, is asked to explain observations from the real world. The learning element, here the discovery architecture described above, is able to evaluate the explanations and "tune" the performance element by changing its model (or theory) of the world.

C.1.1.4 Simultaneous alignment of DNA sequences--MULTAN

Previously, MOLGEN researchers have developed numerous programs to aid in the symbolic analysis of DNA sequences. During the last year Dr. William Bains (a postdoctoral scholar in Professor Kedes' laboratory), completed a program called MULTAN which allows the facile alignment of three or more DNA sequences. This was a major unsolved problem in sequence analysis and the program is now undergoing final testing on the BIONET resource. In the future, we expect that BIONET will support development of application-oriented programs of this type, while MOLGEN and SUMEX will focus on research-oriented systems with major AI goals.

C.2 Research in Progress

We have two major goals over the next several months. The first is to convert and enhance our knowledge-based simulation model within the KEE tool from IntelliCorp, Inc. KEE will be a significant improvement over the Unit System in three areas: speed, functionality, and support. IntelliCorp is providing KEE for use in our research without charge. Studies have indicated that using KEE will unable us to produce a reasonable prototype of our discovery system in about half the time or using the Unit System. Our second goal is to more formally define the learning element of our discovery system and to build a first test system that operates upon the simulation system knowledge base.

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.

5. 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.

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

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

8. 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.

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

10. 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.

11. Friedland, P., and Kedes, L.: *Discovering the secrets of DNA.* (To appear in a joint issue of Communications of the ACM and IEEE/Computer, October, 1985).

12. Friedland, P. and Iwasaki Y.: *The concept and implementation of skeletal plans.* (To appear in Journal of Automated Reasoning, Vol. 1, No. 2, 1985).

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

14. 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.

15. 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.

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

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

18. 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.

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

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

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

E. Funding Support

The MOLGEN grant is titled: MOLGEN: Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing, and Modification. It is NSF Grant MCS-8310236. Current Principal Investigators are Edward A. Feigenbaum Professor of Computer Science and Charles Yanofsky, Professor of Biology. MOLGEN is currently funded from 11/84 to 10/85 at $131,621 including indirect costs as the first year of a three year grant.

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

SUMEX-AIM continues to provide the bulk 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 INTERLISP, TV-EDIT, and BULLETIN BOARD have made possible the project's programming, 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 that SUMEX has begun to evolve into. The ability to utilize LISP workstations for intensive computing while still communicate with all of the other SUMEX resources has been very valuable to our work. We see a satisfactory mode of operation where most programming takes place on the workstations and most electronic communications, information sharing, and document preparation takes place within the mature TOPS-20 environment. The evolution of SUMEX has alleviated most of our previous problems with resource loading and file space. Our current workstations are not quite fast nor sophisticated 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.

The ability for instant communication with a large number of experts in this field has been a determining factor in the success of the MOLGEN project. It has made possible the near instantaneous dissemination of MOLGEN systems to a host of experimental users in laboratories across the country. The wide-ranging input from these users has greatly improved the general utility of our project.

We find it very difficult to find fault with any aspect of the SUMEX resource management. It has made it easy for us to expand our user group, to give demonstrations (through the 20/20 adjunct system as well as the LISP workstations), and to disseminate software to non-SUMEX users overseas.

III. RESEARCH PLANS

A. Project Goals And Plans

Our current work has the following major goals:

1. Use the knowledge base to explain observations that are indeed explainable without changes to the current model. For example, "I have observed a mutation that causes constitutive (uncontrolled) production of tryptophan. How can that be explained within the Jacob-Monod model?" This process will be accomplished by some combination of forward simulation and backward rule-chaining.

2. Begin to recognize when observations are "interesting." Interesting here has one of the following broad meanings:

   a. A seeming direct contradiction to the existing theory.

   b. A statistically rare occurrence (one that is understandable by the current theory, but should not occur very often).

   c. A dramatic confirmation of the existing model.

   d. An observation currently unpredictable by the current model because the model is either not detailed enough or incomplete. The observation in this case must have a relation to the model because an important object of the model is involved or it relates to an effect predicted by the model.

3. Build a mechanism for postulating extensions or corrections to the current theory: a contrained regulatory theory generator. The overall approach to this mechanism is perhaps the most interesting problem in our work. In discussions with other computer scientists, the notion of "or" reasoning where the theory construction process consists of hierarchical refinement of abstract ideas into more detailed ones, and "and" reasoning where the theory is built up in little pieces at many different levels simultaneously has emerged. We see strong evidence for both types of reasoning within Yanofsky's project. In fact, as stated above, the global model of Yanofsky's laboratory is a hybrid one. Individual graduate students performed "and" tasks--filling in details of seemingly unrelated pieces of the model. Yanofsky was the master "or" reasoner, slowly building a hierarchical model of the new regulatory mechanism. It is in this area of our research where the greatest discussion with AI colleagues is needed and which may produce the most significant AI benefits.

4. Build a mechanism for evaluating alternative theories. This would include rating the theories based on plausibility, selectability, completeness, significance, and so on. We hope the evaluation process produces information useful in discriminating among the possible theories.

5. Test the entire structure on the evolving 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, etc.

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.

We support with great enthusiasm the acquisition of satellite computers for technology transfer and hope that the SUMEX staff continues to develop and support these systems. One of the oft-mentioned problems of artificial intelligence research is exactly the problem of taking prototypical systems and applying them to real problems. SUMEX gives the MOLGEN project a chance to conquer that problem and potentially supply scientific computing resources to a national audience of biomedical-medical research scientists.

6.1.3. ONCOCIN Project

ONCOCIN Project

Edward H. Shortliffe, M.D., Ph.D.
Departments of Medicine and Computer Science
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 insure 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 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. With the work currently in its sixth 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 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 have 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.

With daily coordination by the project's data manager, Janice Rohn, the DEC-20 version of ONCOCIN continues to be used on a limited basis in the Stanford Oncology Clinic. The continued dependence on this time-shared computer, however, has prevented us from using ONCOCIN in in many clinical problem areas (other than the lymphomas where clinics are held three mornings per week, and breast cancer where clinic is held one day per week) because of our inability to assure the system's availability with reasonable response time. It is this latter point that has accounted for our decision not to spend a great deal of time developing new protocols to run on the DEC-20 ONCOCIN prototype. Instead we have pressed our effort to adapt ONCOCIN to run on professional workstations which can eventually be dedicated to full time clinic use. We envision these workstations as the model for eventual dissemination of this kind of technology.

In addition to funding from DRR for the workstation conversion effort, we have support from the National Library of Medicine that supports our more basic research activities regarding biomedical knowledge representation, knowledge acquisition, therapy planning, and explanation as it relates to the ONCOCIN task domain. A grant from the NLM to study the therapy planning process was received, and this work (led by Dr. Fagan) is in its second year. This research is investigating how to represent the therapy planning strategies used to decide treatment for patients on the oat cell carcinoma protocol who run into serious problems requiring consultation with the protocol study chairman. Dr. Branimar Sikic, a faculty member from the Stanford University Department of Medicine, and the Study Chairman for the oat cell protocol, is collaborating on this project.

C.2 Research in Progress

The major efforts of the ONCOCIN project over the last year have fallen into three major categories: (1) conversion of ONCOCIN to run on workstations, (2) development of a knowledge acquisition interface (OPAL) for entering new protocols, and (3) research on modeling the strategic therapy selection process (ONYX). Efforts are also in progress to evaluate the system, to document the results of the research, and to disseminate the technology to sites beyond Stanford. We summarize these ongoing research efforts below.

C.2.1 Transfer of the ONCOCIN system from the DEC-20 to the Xerox 1108

In an effort to improve the efficiency of the reimplemented system (and thereby to improve its response time and make it more acceptable to physicians), we have undertaken a substantial system redesign while transferring it to the new machines. An additional commitment in time and programming effort has resulted, but we are confident that the resulting system will be a substantial improvement over the prototype. There have been several aspects to the system's reimplementation during the current year:

- *Reorganization and recoding of existing programs for improved efficiency.* In last year's report, we discussed our first steps in reorganizing the program. A further analysis during the year suggested that we should consider a redesign of the program to take advantage of our experience with the existing program and to respond to advances in artificial intelligence representation methods since ONCOCIN was first designed. In addition, our work during the year on new methods for entering knowledge into the system suggested corresponding improvements in the ways to represent oncologic knowledge in the system (see paper by Musen, et al. for more details on the redesign of the ONCOCIN system).

- *Redesign of the reasoning component.* As a major part of the redesign of the system, we decided to concentrate on methods that would allow for a more efficient search of the knowledge base during the running of a case. We have implemented and are currently debugging a reasoning program that uses a discrimination network to process the cancer protocols. This network allows for a compact representation of information that overlaps elements of multiple protocols, but does not require the program to consider and then disregard information related to protocols that are irrelevant to a particular patient.

- *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 (see the paper by Kahn, et al. for a more detailed description of the temporal network).

- *Extensions to the user interface.* The user interface has been extended so that it can read patient data files of the type that are created by the original ONCOCIN system. This will allow us to transfer currently active patients to the new version of the ONCOCIN system. A detailed description of the user interface is available in the paper by Lane, et al.

- *Connecting the components of the ONCOCIN system.* The reasoning component, user interface, and knowledge acquisition program (described below) have been developed as separate programs. In the final version of the system, the knowledge acquisition program must be able to automatically translate from the graphical input forms into the knowledge base. The reasoner and user interface components are independent programs that run in parallel while communicating with each other. Each of these connections between components has been tested on a limited basis and will continue to be exercised during the next several months.

- *Knowledge engineering tools.* The challenge of coordinating a large software development project, with multiple programmers working in parallel, has necessitated the development of specialized tools to facilitate the process of system construction and maintenance. One area of particular concern has been the need for tools to assist with knowledge base maintenance (see paper by Tsuji and Shortliffe for a discussion of our initial work in this area).

- *System support for the reorganization.* The LISP language that we used to build the first version of ONCOCIN does not explicitly support basic knowledge manipulation techniques (viz. 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 provides the reliability, speed, size, or special memory-manipulation techniques that are needed for our project.

We have accordingly developed a "minimal" object-oriented system to meet these specifications. The object system is currently in use by each component of the new version of ONCOCIN and in the software used to connect the components. In addition, several student projects are now able to use this programming environment.

C.2.2 Interactive Entry of Chemotherapy Protocols by Oncologists (OPAL)

A major effort in this grant year has been the development of software (termed the OPAL system) that will permit physicians who are not computer programmers to enter protocol information into a structured set of forms on a graphical display. Most early expert systems required tedious (and occasionally erroneous) entry of the system's medical knowledge. Each segment of knowledge was transferred from physician to programmer and then entered into the program by the computer expert. Although many programs allowed for specification of a structure within which to organize the information, only minimal attempts were made to define a description that would be generic enough to provide a basis for a series of related knowledge bases in one medical area.

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: multiple protocols (that may be related to each other), experimental research arms in each protocol, drug combinations, individual drugs, and drug modifications. Using the graphically-oriented workstations, this information is presented to the user as computer-generated forms that appear on the screen. As the protocol is described, new forms are added to the computer display to allow for the specification of the special cases that make the protocols so complicated.

Although this design appears to be organized specifically for cancer treatment plans, we believe that the technique 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 could accept *any* possible problem specification.

Using this program we have entered several versions of a small cell lung cancer protocol, and a complicated lymphoma protocol with several different therapies. We are currently implementing the changes suggested by entering these protocols.

C.2.3 Strategic Therapy Planning (ONYX)

As mentioned above, we have begun a new research project to study the therapy planning process, and how strategies which are used to plan therapy in difficult cases might be represented on a computer. This project, which we call the ONYX project, has as its goals: to conduct basic research into the possible representations of the therapy planning process; to develop a computer program to represent this process; and eventually to interface the planning program with ONCOCIN. The project members (Fagan, Tu, Langlotz, and Williams) have spent many hours meeting with Dr. Sikic trying to understand how he plans therapy for patients whose special clinical situation precludes following the standard therapeutic plan described in the protocol document. In March of last year, the group spent two days at Xerox Palo Alto Research Center (PARC), working with Mark Stefik, Daniel Bobrow and Sanjay Mittal of PARC on possible representations for the knowledge structures and how such a program might run using the LOOPS knowledge programming system. A prototype version of this program is currently being tested. The prototype program has been designed as two components: the strategic planning program and the qualitative simulation builder. The strategic planning program is capable of turning the patient's medical data and knowledge of the intent of the protocol into a small number of plausible protocol modifications for the current point in time, and conditional modifications for the near future. Another component of the system is capable of building simulation models using the graphical abilities of the 1108 workstation. The first test of this component is the construction of a model of the effects of chemotherapy drugs on the bone marrow of the patient. During the next year of research this type of qualitative simulation model will be integrated into the strategic planning program.

C.2.4 Evaluations of ONCOCIN's performance

We have completed our first three formal studies of ONCOCIN's DEC-20 version (see papers by Kent et al. and Hickam et al. for results of two of these; written reports on the third is in preparation). Lessons learned in these initial studies have led to revisions both in the design of ONCOCIN and in our plans for evaluation studies of the 1108 version of the system when it is implemented at non-Stanford sites in later years.

C.2.5 Documentation

We have developed a videotape that discusses and demonstrates our research on the workstation version of our system. This tape has been shown at national meetings and has been extensively distributed to researchers internationally who have shown an interest in our work. The publication list that accompanies this report further documents the design decisions we have made in developing the new version of ONCOCIN.

C.2.6 Dissemination

In anticipation of completion of the workstation version of ONCOCIN, we are beginning to plan for an experiment in which we will install ONCOCIN workstations in private oncology offices in San Jose and Fresno. An application proposing this work is current under review.

D. Publications Since January 1984

1. (*) Buchanan, B.G. and Shortliffe, E.H.: *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA., 1984. [book]

2. (*) Clancey, W.J. and Shortliffe, E.H.: *Readings in Medical Artificial Intelligence: The First Decade.* Addison-Wesley, Reading, MA., 1984. [book]

3. Clancey, W.J. and Shortliffe, E.H.: *Strategies for medical knowledge engineering: Lessons from the first decade.* To appear in the Proceedings of the AAMSI Congress 85, San Francisco, CA., May 1985.

4. Differding, J.C.: *The OPAL interface: General Overview.* Working paper. August 1984.

5. (*) Fagan, L.: *New Directions for Expert Systems: Examples from the ONCOCIN Project.* To appear in the Proceedings of AAMSI Congress 85, San Francisco, CA., May 1985.

6. (*) 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.* Submitted for publication, May 1985.

7. (*) Kahn, M.G., Ferguson, J., Shortliffe, E.H., Fagan, L.: *An approach for structuring temporal information in the ONCOCIN system.* To appear in the

Proceedings of the Symposium on Computer Applications in Medical Care, Baltimore, MD., November 1985.

8. (*) 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*. Submitted for publication, March 1985.

9. (*) Lane, C.D., Differding, J.C., Shortliffe, E.H.: *Design of a graphic interface for a medical expert system*. (Memo KSL-85-15). Working paper.

10. (*) Langlotz, C., Fagan, L., Tu, S., Williams, J., Sikic, B.: *ONYX: An architecture for planning in uncertain environments*. To appear in the Proceedings of International Joint Conference on Artificial Intelligence, Los Angeles, CA., August 1985.

11. (*) Langlotz, C.P. and Shortliffe, E.H.: *Adapting a consultation system to critique user plans*. In Developments in Expert Systems, (M. Coombs, ed.), pp. 77-94, London: Academic Press, 1984.

12. (*) Musen, M., Langlotz, C., Fagan, L., Shortliffe, E.H.: *Rationale for knowledge base redesign in a medical advice system*. To appear in the Proceedings of AAMSI Congress 85, San Francisco, CA., May 1985.

13. Shortliffe, E.H.: *The science of biomedical computing*.Medical Informatics, Vol.9, Nos. 3/4, 185-193 (1984).

14. (*) Shortliffe, E.H.:*Reasoning methods in medical consultation systems: artificial intelligence approaches* (tutorial). Computer Programs in Biomedicine 18:5-14 (1984).

15. Shortliffe, E. H.: *Explanation capabilities for medical consultation systems* (tutorial). Proceedings of AAMSI Congress 84 (D. Lindberg and M. Collen, Eds.), pp. 193-197, San Francisco, May 1984.

16. Shortliffe, E.H. and Fagan, L.M.: *Artificial intelligence: the expert systems approach to medical consultation*. Proceedings of the 6th Annual International Symposium on Computers in Critical Care and Pulmonary Medicine, Heidelberg, Germany, June 1984.

17. (*) Shortliffe, E.H.: *Update on ONCOCIN: A chemotherapy advisor for clinical oncology*. Proceedings of the Symposium on Computer Applications in Medical Care, November 1984.

18. (*) Tsuji, S. and Shortliffe, E.H.: *Graphics for knowledge engineers: a window on knowledge base management* (Memo KSL-85-11). Submitted for publication, April 1985.

E. Funding Support

Grant Title: "Studies in the Dissemination of Consultation Systems"
Principal Investigator: Edward H. Shortliffe
Agency: Biomedical Research Technology Program, Division of Research Resources
ID Number: RR 01613
Term: July 1983 to June 1986
Total award: $624,455

Current award: (7/84-6/85): $222,511 (Direct costs)

Grant Title: "Therapy-planning strategies for consultation by computer"
Principal Investigator: Edward H. Shortliffe
Agency: National Library of Medicine
ID Number: LM-04136
Term: August 1983 to July 1986
Total award: $211,851
Current award: (8/84-7/85) $69,875 (Direct costs)

Grant Title: "Postdoctoral Training in Medical Information Science"
Principal Investigator: Edward H. Shortliffe
Agency: National Library of Medicine
ID Number: 1 T32 LM07033
Term: July 1, 1984 - June 30, 1989
Total award: $903,718
Current award: (7/84-6/85) $79,059 (Direct costs)

Grant Title: Explanation of Computer-Assisted Therapy Plans"
Principal Investigator: Lawrence M. Fagan
Agency: National Library of Medicine (New Investigator Grant)
ID Number: 1 R23 LM04316
Term: February 1985-January 1988
Total award: $107,441
Current award: (2/85-1/86) $37,500 (Direct Costs)

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 1986, renewable until June 1988
Total award: $150,000 ($50,000 annually).

Grant Title: Information structure and use in knowledge-based expert systems
Principal Investigator: Bruce G. Buchanan
Co-Principal Investigator: Edward H. Shortliffe
Agency: National Science Foundation - IST83-12148
Term: March 1, 1984 - February 28, 1987
Total award: $330,000 (includes indirects)

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 (both the prototype system running in the clinic and the newer work transferring the system to professional workstations). We also demonstrated our developing workstation code in the Xerox exhibit in the trade show associated with AAAI-84 in Austin, Texas. 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*, whereas ONCOCIN has been featured on the "NBC Nightly News", the PBS

"Health Notes" series, and "The MacNeil-Lehrer Report." Due to the frequent requests for ONCOCIN demonstrations, we have produced a videotape about the ONCOCIN research which includes demonstrations of our the professional workstation research projects and the 2020-based clinic system. The tape has been shown at several national meetings, including the 1984 Workshop on Artificial Intelligence in Medicine, the 1984 meeting of the Society for Medical Decision Making, and the 1985 meeting of the Society for Research and Education in Primary Care Internal Medicine. The tape has also been shown to both national and international researchers in biomedical computing.

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 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 start out 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 on in July of 1983 in *Computers and Biomedical Research.*

B. Sharing and Interaction with Other SUMEX-AIM Projects

The community created on the SUMEX resource has other benefits that 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 exchange 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 on opposite coasts of the country.

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 hard 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.

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 transfer the oncology prototype from its current research computer to a professional workstation that provides a model for cost-effective dissemination of clinical consultation systems.* To meet this specific aim we will we will continue the basic and applied programming efforts (ONCOCIN, OPAL, and ONYX) described earlier in this report.

2. *To encode and implement for use by ONCOCIN the commonly used chemotherapy protocols from our oncology clinic.* In the coming year, we will:

    - Complete our OPAL protocol entry system

    - Continue entry of additional protocols, hopefully at the rate of one protocol/month (including testing)

    - Place a version of the OPAL protocol entry system into the clinic for use by physicians as a graphical reference guide to the protocols.

3. *To introduce ONCOCIN gradually for ongoing use so that by mid-1986 two professional workstations will be available in the oncology clinic to assist in the management of cancer patients.* During the next year, we will:

    - Implement the first workstation-based ONCOCIN system for use by physicians in the oncology clinic by the end of the calendar year 1985, adding a second workstation within a few months thereafter

    - Continue to operate the DEC-2020 version to maintain continuity of support in the clinic setting until the workstation version is fully operational.

B. Justification and Requirements for Continued SUMEX Use

All the work we are doing (ONCOCIN plus continued use of the original MYCIN program) continues to be dependent on daily use of the SUMEX resource. Although much of the ONCOCIN work is shifting to Xerox workstations, the SUMEX 2060 and the 2020 continue to be key elements in our research plan. The programs all make assumptions regarding the computing environment in which they operate, and the ONCOCIN prototype currently used in the clinic depends upon proximity to the DEC 2020 which enables us to use a 9600 baud interface.

In addition, we have long appreciated the benefits of GUEST and network access to the programs we are developing. SUMEX greatly enhances our ability to obtain feedback from interested physicians and computer scientists around the country. Network access has also permitted high quality formal demonstrations of our work both from around the United States and from sites abroad (e.g., Finland, Japan, Sweden, Switzerland).

The main development of our project will continue to take place on Dandelion lisp machines that we have purchased or have been donated by XEROX corporation. We also have special needs for more computing power for our ONYX therapy planning research, and have been able to share an upgraded Dandelion loaned by SUMEX for this work.

C. Requirements for Additional Computing Resources

The acquisition of the DEC 2020 by SUMEX was crucial to the growth of our research work. It has insured high quality demonstrations and has enabled us to develop a system (ONCOCIN) for real-world use in a clinical setting. As we have begun to develop systems that are potentially useful as stand-alone packages (i.e., an exportable

ONCOCIN), the addition of personal workstations has provided particularly valuable new resources. We have made a commitment to the smaller Interlisp-D machines (Dandelions) produced by Xerox, and our work will increasingly transfer to them over the next several years. Our current funding supports our effort to implement ONCOCIN on workstations in the Stanford oncology clinic (and eventually to move the program to non-Stanford environments) but we will simultaneously continue to require access to Interlisp on upgraded workstations for extremely CPU intensive tasks. Although our dependence on SUMEX for workstations has decreased due to a recent gift from XEROX, our requirements for network support of the machines has drastically increased. Individual machines do not provide sufficient space to store all of the software used in our project, nor to provide backup or long term storage of work in progress. It is the networks, file storage devices, protocol converters, and other parts of the SUMEX network that hold our project together. In addition, with a research group of about 20 people, we are taking advantage of file sharing, electronic mail, and other information coordinating activities provided by the DEC 2060. We hope that with systems support and research by SUMEX staff, we will be able to gradually move away from a need for the central coordinating machine over the next five years.

The acquisition of the DEC 2060, coupled with our increasing use of workstations, has greatly helped with the problems in SUMEX response time that we had described in previous annual reports. We are extremely grateful for access both to the central machine and to the research workstations on which we are currently building the new ONCOCIN prototype. The D-machine's address space is permitting development of the large knowledge base that ONCOCIN requires. The graphics capability of the workstations has also enabled us to develop new methods for presenting material to naive users. In addition, the D-machines have provided a reliable, constant "load-average" machine for running experiments with physicians and doing development work. The development of ONCOCIN on the Dandelion will demonstrate the feasibility of running intelligent consultation systems on small, affordable machines in physicians' offices and other remote sites.

D. Recommendations for Future Community and Resource Development

SUMEX is providing an excellent research environment and we are delighted with the help that SUMEX staff have provided implementing enhanced system features on the 2060 and on the workstations. We feel that we have a highly acceptable research environment in which to undertake our work. Workstation availability is becoming increasingly crucial to our research, and we have found over the past year that workstation access is at a premium. The SUMEX staff has been very helpful and understanding about our needs for workstation access, allowing us Dandelion use wherever possible, and providing us with systems-level support when needed. We look forward to the arrival of additional advanced workstations and the development of a more distributed computing environment through SUMEX-AIM.

6.1.4. 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 (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 [34].

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 speed up the determination.

C. Highlights of Progress

We have constructed a prototype of such a program, called PROTEAN, designed on the blackboard model [16], [26]. It is implemented in BB1 [27], a framework system for building blackboard systems that control their own problem-solving behavior [28](see discussion of BB1 above). We have coupled the reasoning program with an IRIS graphics terminal (shared with SUMEX) which displays protein structures at different levels of detail. This provides a visual understanding of how the program is behaving, which is essential for this problem.

PROTEAN embodies the following experimental techniques for coping with the complexities of constraint satisfaction:

1. The problem-solver partitions each problem into a network of loosely-coupled sub-problems. PROTEAN partitions the problem of positioning all of a protein's constituent structures within a global coordinate system into sub-problems of positioning individual pieces of structures and their immediate neighbors within local coordinate systems. It subsequently composes the most constrained partial solutions developed for these sub-problems in a complete solution for the entire protein. This partitioning and composition technique reduces the combinatorics of search. It also

introduces additional constraints in the global characteristics of internally constrained partial solutions. For example, the conformations of partial protein solutions constrain their composability with other partial solutions.

2. The problem-solver attempts to solve sub-problems and coordinate solutions at multiple levels of abstraction, where lower levels of abstraction partition solution elements with finer granularity. For example, PROTEAN operates at three levels of abstraction. At the "Solid" level, it positions elements of the protein's secondary structure: alpha-helices, beta-sheets, and random coils. At the "Blob" level, it positions elements of the protein's primary structure of amino acids: peptide units and side-chains. At the "Atom" level, it positions the protein's individual atoms. Partial solutions at higher levels of abstraction reduce the combinatorics of search at lower levels. Conversely, tightly constrained partial solutions at lower levels introduce new constraints on higher-level solutions.

3. The problem-solver forbears hypothesizing specific partial solutions for a sub-problem in favor of preserving the "family" of solutions consistent with all constraints applied thus far. For example, in positioning a helix within a partial solution, PROTEAN does not attempt to identify a unique spatial position for the helix. Instead, it identifies the entire spatial volume within which the helix might lie, given the constraints applied thus far. Preserving the family of legal solutions accommodates problems with incomplete constraints; the solution is only as constrained as the data are constraining. It also accommodates incompatible constraints by permitting disjunctive sub-families. For PROTEAN, disjunctive sub-volumes imply that the associated structure lies within any one of the sub-volumes or, if the structure is mobile, that it may move from one sub-volume to another.

4. The problem-solver applies constraints one at a time, successively restricting the family of solutions hypothesized for different sub-problems. PROTEAN successively applies constraints on the positions of protein structures, successively restricting the spatial volumes within which they may lie. Independent application of different constraints finesses the problem of integrating qualitatively different kinds of constraints by simply integrating their results. In addition, successive restriction of the family of solutions obviates guessing which specific solutions within a family are likely to be consistent with subsequently applied constraints and the otherwise inevitable back-tracking.

5. The problem-solver tolerates overlapping solutions for different sub-problems. For example, in identifying the volume within which structure-a might lie in partial solution 1, PROTEAN may include part of the volume identified for structure-b. Toleration of overlapping partial solutions is another accommodation of incomplete or incompatible constraints and potentially dynamic solutions. For PROTEAN, overlapping volumes for two protein structures indicate either: (a) that the two structures actually occupy disjoint sub-volumes that cannot be distinguished within the larger, overlapping volumes identified for them because the constraints are incomplete; or (b) that the two structures are mobile and alternately occupy the shared volume.

6. The problem-solver reasons explicitly about control of its own problem-solving actions: which sub-problems it will attack, which partial solutions it will expand, and which constraints it will apply. Control reasoning guides the problem-solver to perform actions that minimize computation, while maximizing progress toward a complete solution (see section 3.2.1). It also

provides a foundation for the problem-solver's explanation of problem-solving activities and intermediate partial solutions (see section 3.2.2) and for its learning of new control heuristics (see section 5.5).

The current version of PROTEAN has six knowledge sources that demonstrate the reasoning techniques described above. These knowledge sources develop partial solutions that position multiple helices at the Solid level and refine those helices at the Blob level. Proposed work will introduce knowledge sources that operate on other protein structures at the Solid level, as well as knowledge sources that apply the reasoning techniques at the Blob and Atom levels. We also will investigate emergent constraints entailed in reliable partial solutions, composition of partial solutions into complete solutions, and intelligent control.

D. Relevant Publications

1. Erman, L.D., Hayes-Roth, B., Lesser, V.R., Reddy, D.R.:*The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty*. ACM Computing Surveys 12(2):213-254, June, 1980.

2. Hayes-Roth, B.: *The Blackboard Architecture: A General Framework for Problem Solving?* Report HPP-83-30, Department of Computer Science, Stanford University, 1983.

3. 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.

4. Hayes-Roth, B.:*A Blackboard Architecture for Control*. Artificial Intelligence In Press, 1985.

5. Hayes-Roth, B. and Hewett, M.: *Learning Control Heuristics in BB1*. Report HPP-85-2, Department of Computer Science, 1985.

6. 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.

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

Total Amount: $100,000

Dates: Nov 1, 1984/Oct 31 1986

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

Robert Langridge was visiting at Stanford last year, and informal discussions with him and his group have continued in this year.

C. Critique of Resource Management

The SUMEX staff has continued to be most cooperative in getting this project started. Without their persistence, we would not have been able to obtain Ethernet software for the IRIS graphics terminal from Xerox.

III. RESEARCH PLANS

A. Goals & Plans

Our long-range goal is to build an automatic interpretation system similar to CRYSALIS (which worked with x-ray crystallography data). In the shorter term, we are building interactive programs that aid in the interpretation of NMR data on small proteins. The current version of PROTEAN has six knowledge sources that demonstrate the reasoning techniques described above. These knowledge sources develop partial solutions that position multiple helices at the Solid level and refine those helices at the Blob level. The proposed research would expand PROTEAN to include knowledge sources that:

1. construct partial solutions combining helices, beta sheets, and random coils at the Solid level;
2. merge highly constrained partial solutions at the Solid level;
3. refine Solid level solutions in terms of the relative positions of constituent peptide units and side chains at the Blob level;
4. further restrict the relative locations of peptide units and side chains relative to one another at the Blob level;
5. propagate emergent constraints at the Blob level back up to the Solid level to further restrict the relative positions of superordinate helices, beta sheets, and random coils;
6. refine Blob level solutions at the Atom level;
7. further restrict the relative locations of atoms relative to one another;
8. propagate emergent constraints at the Atom level back up to the Blob level to further restrict the relative positions of superordinate peptide units and side chains.

The research will also develop a set of control knowledge sources to guide PROTEAN's application of constraints to identify the family of legal protein conformations as efficiently as possible. And we expect to improve the graphics interface to provide more functionality and options for viewing partial structures.

B. Justification for continued SUMEX use

We will continue to use SUMEX for developing parts of the program before integrating them with the whole system. We are using Interlisp to implement the Blackboard model and knowledge structures most flexibly and quickly.

C. Need for other computing resources

In this stage of development we need more computer cycles and hope to have access to additional D-machines. We expect to upgrade the Silicon Graphics IRIS terminal to a workstation for more efficiency in the subprograms doing computational geometry.

6.1.5. RADIX Project

The RADIX Project: Deriving Medical Knowledge from Time-Oriented Clinical Databases

Robert L. Blum, M.D., Ph.D.
Department of Computer Science
Stanford University

Gio C. M. Wiederhold, Ph.D.
Departments of Computer Science and Medicine
Stanford University

I. SUMMARY OF RESEARCH PROGRAM

A. Technical Goals - Introduction

Medical and Computer Science Goals -- The long-range objectives of our project, called RADIX (formerly RX), are 1) to increase the validity of medical knowledge derived from large time-oriented databases containing routine, non-randomized clinical data, 2) to provide knowledgeable assistance to a research investigator in studying medical hypotheses on large databases, 3) to fully automate the process of hypothesis generation and exploratory confirmation. For system development we have used a subset of the ARAMIS database.

Computerized clinical databases and automated medical records systems have been under development throughout the world for at least a decade. Among the earliest of these endeavors was 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. The ARAMIS Project has now been generalized to include databases for many chronic diseases other than arthritis.

The 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 RADIX computer program is designed to examine a time-oriented clinical database such as ARAMIS and to produce a set of (possibly) causal relationships. The algorithm exploits three properties of causal relationships: time precedence, correlation, and nonspuriousness. First, a Discovery Module uses lagged, nonparametric correlations to generate an ordered list of tentative relationships. Second, a Study Module uses a knowledge base (KB) of medicine and statistics to try to establish nonspuriousness by controlling for known confounders.

The principal innovations of RADIX are the Study Module and the KB. The Study

Module takes a causal hypothesis obtained from the Discovery Module and produces a comprehensive study design, using knowledge from the KB. The study design is then executed by an on-line statistical package, and the results are automatically incorporated into the KB. Each new causal relationship is incorporated as a machine-readable record specifying its intensity, distribution across patients, functional form, clinical setting, validity, and evidence. In determining the confounders of a new hypothesis the Study Module uses previously "learned" causal relationships.

In creating a study design the Study Module follows accepted principles of epidemiological research. It determines study feasibility and study design: cross-sectional versus longitudinal. It uses the KB to determine the confounders of a given hypothesis, and it selects methods for controlling their influence: elimination of patient records, elimination of confounding time intervals, or statistical control. The Study Module then determines an appropriate statistical method, using knowledge stored as production rules. Most studies have used a longitudinal design involving a multiple regression model applied to individual patient records. Results across patients are combined using weights based on the precision of the estimated regression coefficient for each patient.

B. Medical Relevance and Collaboration

As a test bed for system development our focus of attention has been on 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.

In the future we anticipate possible collaborations with other project users of the TOD System such as the National Stroke Data Bank, the Northern California Oncology Group, and the Stanford Divisions of Oncology and of Radiation Therapy.

We believe that this research project is broadly applicable to the entire gamut of chronic diseases that constitute the bulk of morbidity and mortality in the United States. Consider five major diagnostic categories responsible for approximately two thirds of the two million deaths per year in the United States: myocardial infarction, stroke, cancer, hypertension, and diabetes. Therapy for each of these diagnoses is fraught with controversy concerning the balance of benefits versus costs.

1. Myocardial Infarction: Indications for and efficacy of coronary artery bypass graft vs. medical management alone. Indications for long-term antiarrhythmics ... long-term anticoagulants. Benefits of cholesterol-lowering diets, exercise, etc.

2. Stroke: Efficacy of long-term anti-platelet agents, long-term anticoagulation. Indications for revascularization.

3. Cancer: Relative efficacy of radiation therapy, chemotherapy, surgical excision - singly or in combination. Optimal frequency of screening procedures. Prophylactic therapy.

4. Hypertension: Indications for therapy. Efficacy versus adverse effects of chronic antihypertensive drugs. Role of various diagnostic tests such as renal arteriography in work-up.

5. Diabetes: Influence of insulin administration on microvascular complications. Role of oral hypoglycemics.

Despite the expenditure of billions of dollars over recent years for randomized controlled trials (RCT's) designed to answer these and other questions, answers have been slow in coming. RCT's are expensive in terms of funds and personnel. The therapeutic questions in clinical medicine are too numerous for each to be addressed by its own series of RCT's.

On the other hand, the data regularly gathered in patient records in the course of the normal performance of health care delivery are a rich and largely underutilized resource. The ease of accessibility and manipulation of these data afforded by computerized clinical databases holds out the possibility of a major new resource for acquiring knowledge on the evolution and therapy of chronic diseases.

The goal of the research that we are pursuing on SUMEX is to increase the reliability of knowledge derived from clinical data banks with the hope of providing a new tool for augmenting knowledge of diseases and therapies as a supplement to knowledge derived from formal prospective clinical trials. Furthermore, the incorporation of knowledge from both clinical data banks and other sources into a uniform knowledge base should increase the ease of access by individual clinicians to this knowledge and thereby facilitate both the practice of medicine as well as the investigation of human disease processes.

C. Highlights of Research Progress

C.1 April 1984 to April 1985

Our primary accomplishments in this period have been the following:

1) completion of modifications to RADIX to accommodate the one hundred-fold increase in the size of our database to 1700 patients,

2) carrying out and publishing the study of the effect of prednisone on serum cholesterol on this expanded database,

3) publishing a description of the two-stage regression method adapted by us to this study,

4) completion of a System Programmer's Manuals and User's Manual

5) initiation of transfer of RADIX to Xerox 1108 personal work stations.

C.1.1 Modifications to RADIX for the enlarged database

Extensive modifications to RADIX were required to deal with the 100-fold increase in the size of the database. The modifications necessary to run the study module automatically on the prednisone/cholesterol study were completed this year.

C.1.2 Prednisone/chlosterol study on enlarged database

We have carried out the automated study of the effect of prednisone on serum cholesterol using the new 1700 patient database. It has strongly confirmed the effect previously observed in the 50-patient SLE database. In addition, we are examining the effect in non-SLE patients and in other patient subsets. We are also examining alternative pharmacokinetic models for the prednione effect using the newly available data.

An extensive paper describing the RADIX System and reporting the results of the prednisone/cholesterol study has been submitted to a major medical journal for publication.

C.1.3 Publish description of 2-stage regression method

A detailed description of the 2-stage regression method used by us for the above study has been sent to a major statistical journal for publication.

C.1.4 Documentation

A two-volume System Programmer's Manual and a User's Manual describing implementation, maintenance and use of the system at Stanford has been completed. In addition, a complete set of the files needed for on-line demonstrations has been created, separating them from the working versions.

C.1.5 Transer of RADIX to D-Machines

Preliminary work on implementing RADIX on D-Machines has begun. This will continue in coming years.

C.1.6 Other accomplishments

We have presented the results of our research at several conferences during the year. Additional publications for the year are noted in the section on publications.

In addition, new work on the theory of medical knowledge representation is described below.

C.2 Research in Progress

Our current work is focusing on problems involved in the representation of medical knowledge. Specifically, we are developing new methods for representing medical causal relationships. These have been represented in most other systems as simply binary relationships with conditional probabilities or certainty factors. In our project we are exploring the representation of causal relationships using categorical, rank, and real-valued relationships, as well as binary ones. We anticipate that these relationships will a) lend greater accuracy to predictions and diagnoses made by medical consultation systems, and b) will enable medical knowledge bases to be more compact and perspicuous.

In addition to this theoretical work, we are also pursuing two applications. First, we are developing a system for using a medical knowledge base to summarize a patient's time-oriented record. That is, our intended system will take as input a table of signs, symptoms, and lab values of the patient over time and will transform this into a time-oriented summary of arbitrary detail. This application draws upon our existing work in representation of causal relationships and in labeling time-oriented records.

Our second application involves the development of methods for automating the discovery of new relationships from time-oriented patient records. Here, we have elaborated a number of methods that we intend to exploit in a newly designed version of our discovery module. These methods take advantage of pre-existing medical knowledge by using analogical reasoning. We expect that this work will be facilitated by our recent acquisition of the KEE knowledge representation system, courtesy of Intellicorp, for use on our Xerox 1108's.

D. Publications

1. Blum, R.L.: *Two Stage Regression: Application to a Time-Oriented Clinical Database.* (Submitted for publication to the Journal of Statistics in Medicine.)

2. Blum, R.L.: *Prednisone Elevates Cholesterol: An Automated Study of Longitudinal Clinical Data.* (Submitted to the Annals of Internal Medicine.)

3. Blum, R.L., and Walker, M.G.: *Minimycin: A Miniature Rule-Based System* (Accepted for publication by M.D.Computing)

4. Blum, R.L.: *Modeling and encoding clinical causal relationships.* Proceedings of SCAMC, Baltimore, MD, October, 1983.

5. Blum, R.L.: *Representation of empirically derived causal relationships.* IJCAI, Karlsruhe, West Germany, August, 1983 .

6. Blum, R.L.: *Machine representation of clinical causal relationships.* MEDINFO 83, Amsterdam, August, 1983.

7. 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.

8. 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.

9. Blum, R.L.: *Induction of causal relationships from a time-oriented clinical database: An overview of the RX project.* Proc. AAAI, Pittsburgh, August, 1982.

10. Blum, R.L.: *Automated induction of causal relationships from a time-oriented clinical database: The RX project.* Proc. AMIA San Francisco, 1982.

11. 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.

12. 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.

13. 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.

14. Blum, R.L.: *Displaying clinical data from a time-oriented database.* Computers in Biol. and Med. 11(4):197-210, 1981.

15. 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)

16. 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.

17. Blum, R.L.: *The RX project: A medical consultation system integrating clinical data banking and artificial intelligence methodologies*, Stanford University Ph.D. thesis proposal, August, 1978.

18. 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.

19. Walker, M.G., and Blum, R.L.: *A Lisp Tutorial.* (Submitted for publication to M.D.Computing.)

20. Wiederhold, G.: *Knowledge and Database Management*, IEEE Software Premier Issue, Jan.1984, pp.63--73.

21. 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.

22. 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.

23. Wiederhold, G.: IN D.A.B. Lindberg and P.L. Reichertz (Eds.), *Databases for Health Care*, Lecture Notes in Medical Informatics, Springer-Verlag, 1981.

24. Wiederhold, G.: *Database technology in health care.* J. Medical Systems 5(3):175-196, 1981.

E. Funding Support Status

1) Representation and Use of Causal Knowledge for Inference from Databases
Robert L. Blum, M.D., Ph.D.: Principal Investigator
National Science Foundation: IST 83-17858
Total award: $89,597 (direct + indirect)
Term: March 15, 1984 through March 14, 1986

2) Deriving Knowledge from Clinical Databases
Gio C. M. Wiederhold, Ph.D.: Principal Investigator
National Library of Medicine: LM-04334
Total award: $291,192 (direct)
Term: May 1, 1984 through November 30, 1986

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Collaborations

During the past year we completed System Programmer's Manuals and a User's Manual as steps towards making the system available to outside collaborators. Once the RADIX program is developed, we would anticipate collaboration with some of the ARAMIS project sites in the further development of a knowledge base pertaining to the chronic arthritides. The ARAMIS Project at the Stanford Center for Information Technology is used by a number of institutions around the country via commercial leased lines to store and process their data. These institutions include the University of California School of Medicine, San Francisco and Los Angeles; The Phoenix Arthritis Center, Phoenix; The University of Cincinnati School of Medicine; The University of Pittsburgh School of Medicine; Kansas University; and The University of Saskatchewan. All of the rheumatologists at these sites have closely collaborated with the development of ARAMIS, and their interest in and use of the RADIX project is anticipated. We hasten to mention that we do not expect SUMEX to support the active use of RADIX as an on-going service to this extensive network of arthritis centers, but we would like to be able to allow the national centers to participate in the development of the arthritis knowledge base and to test that knowledge base on their own clinical data banks.

B. Interactions with Other SUMEX-AIM Projects

This past year, in moving our work to the Xerox 1108's, we have had frequent consultations with members of the Oncocin staff and have made use of several utility programs developed by them including hash file facilities and programs facilitating the tabular display of data.

Regular communication on programming details is facilitated by the on-line mail system.

C. Critique of Resource Management

The DEC System 20 continues to provide acceptable performance, but it is frequently heavily loaded at peek hours.

The SUMEX resource management continues to be accessible and and quite helpful.

III. RESEARCH PLANS

A. Project Goals and Plans

The overall goal of the RADIX Project is to develop a computerized medical information system capable of accurately extracting medical knowledge pertaining to the therapy and evolution of chronic diseases from a database consisting of a collection of stored patient records.

SHORT-TERM GOALS --

For the past two years we have concentrated principally on publishing and presenting our earlier AI results, on acquisition of a 1700 patient database, on medical studies based on the enlarged database, and on reporting the medical results and statistical techniques arising from our research. This is in concert with the long-term goal of ensuring that the work of the SUMEX / Artificial Intelligence in Medicine community be disseminated and applied in the general medical community.

During the coming two years we will concentrate much more on the artificial intelligence aspects of RADIX. We were successful last year in obtaining funding from the National Library of Medicine and the National Science Foundation to pursue this work. In particular, we will be deeply concerned with the representation of causal, temporal, and quantitative medical knowledge. It has become clear that these types of knowledge are crucial for the RADIX tasks of automated discovery of medical knowledge and the provision of intelligent automated assistance to clinical researchers, in addition to their generally perceived value in other medical expert systems applications.

LONG-RANGE GOALS -- There are two inter-related long-range goals of the RADIX Project: 1) automatic discovery of knowledge in a large time-oriented database and 2) provision of assistance to a clinician who is interested in testing a specific hypothesis. These tasks overlap to the extent that some of the algorithms used for discovery are also used in the process of testing an hypothesis.

We hope to make these algorithms sufficiently robust that they will work over a broad range of hypotheses and over a broad spectrum of data distributions in the patient records.

B. Justification and Requirements for Continued Use of SUMEX

Computerized clinical data banks possess great potential as tools for assessing the efficacy of new diagnostic and therapeutic modalities, for monitoring the quality of health care delivery, and for support of basic medical research. Because of this potential, many clinical data banks have recently been developed throughout the United States. However, once the initial problems of data acquisition, storage, and retrieval have been dealt with, there remains a set of complex problems inherent in the task of accurately inferring medical knowledge from a collection of observations in patient records. These problems concern the complexity of disease and outcome definitions, the complexity of time relationships, potential biases in compared subsets, and missing and outlying data. The major problem of medical data banking is in the reliable inference of medical knowledge from primary observational data.

We see in the RADIX Project a method of solution to this problem through the utilization of knowledge engineering techniques from artificial intelligence. The RADIX Project, in providing this solution, will provide an important conceptual and technological link to a large community of medical research groups involved in the treatment and study of the chronic arthritides throughout the United States and Canada, who are presently using the ARAMIS Data Bank through the CIT facility via TELENET.

Beyond the arthritis centers which we have mentioned in this report, the TOD (Time-Oriented Data Base) User Group involves a broad range of university and community medical institutions involved in the treatment of cancer, stroke, cardiovascular disease, nephrologic disease, and others. Through the RADIX Project, the opportunity will be provided to foster national collaborations with these research groups and to provide a major arena in which to demonstrate the utility of artificial intelligence to clinical medicine.

C. Recommendations for Resource Development

The on-going acquisition of personal work-station Lisp processors is a very positive step, as these provide an excellent environment for program development, and can serve as a vehicle for providing programs to collaborators at other sites. Continued acquisitions are very desirable.

We also would hope that the central SUMEX facility, the DEC 2060, would continue to be supported. We continue to make constant use of this machine for text-editing, document preparation, file and database handling, communications, and program demos.

6.2. 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.

6.2.1. CADUCEUS Project

CADUCEUS Project

J. D. Myers, M.D. and Harry E. Pople, Jr., Ph.D.
University of Pittsburgh
Decision Systems Laboratory
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-1, 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-1 has performed at the level of the skilled internist, but the experience has high-lighted several areas for improvement.

B. Medical Relevance and Collaboration

The program inherently has direct and substantial medical relevance.

The institution of collaborative studies with other institutions has been deferred pending completion of the programs and knowledge base enhancements required for CADUCEUS. The installation of our own, dedicated VAX computer can be expected to aid considerably any future collaboration.

The INTERNIST-1 program has been used in recent years to develop patient management problems for the American College of Physician's Medical Knowledge Self-assessment Program, and to develop patient management problems and test cases for the Part III Examination and the developing computerized testing program of the National Board of Medical Examiners. In addition, selected other medical schools are employing the INTERNIST-1 knowledge base for medical student and house staff education.

----Accomplishments this past year

During 1983-84, under the supervision of Drs. Miller and Myers, Dr. Michael First, a former University of Pittsburgh medical student with extensive experience working in the Decision Systems Laboratory, developed a program called QUICK (QUick Index into Caduceus Knowledge), a prototypical electronic textbook of medicine utilizing the INTERNIST-1 knowledge base as its foundation. A paper describing QUICK, including an informal trial evaluating its utility, appears in the April 1985 issue of Computers and Biomedical Research. The residents in Internal Medicine who were given access to QUICK rated it favorably as a source of medical information. All three hospitals participating in the evaluation of QUICK have requested that they be given continued access to the program. An effort is being made to adapt QUICK to the IBM-PC for easier use by physicians.

From 1981 through 1983, Dr. Miller, under NLM New Investigator Award 5R23-LM03589, developed a clinical patient case simulator program, CPCS. The goal of the project was to build a program and knowledge base capable of constructing, de novo, logically consistent and clinically plausible artificial patient case summaries. Such a program would be useful in helping medical students to broaden their diagnostic skills. The program might also be used in generating cases for testing purposes, as this is now done manually by the National Board of Medical Examiners for their certification examinations. CPCS was a successful feasibility study; its performance has not yet been formally evaluated. Plans have been made to convert the entire INTERNIST-1 knowledge base into the format used by CPCS, and to add a better representation of time to the CPCS program and knowledge base.

Drs. Miller and Myers have developed, as part of the CPCS project, a new format for the internal medicine knowledge base. The specific details of this format have been described in previous progress reports. We have, in a period of three to four man-months, converted on paper the INTERNIST-1 knowledge base for liver diseases into the new format. This represents about one-sixth of the entire INTERNIST-1 knowledge base.

Dr. Miller has written an editor program to enter and maintain the new knowledge base, using Franz Lisp. At present, that editor program has been used to construct some 15-17 diagnoses from the INTERNIST-1 liver diseases. This includes creation of some 50-70 facets describing the underlying pathophysiology. A total of 200-300 findings have been entered into the new knowledge base, and because of their complexity, they correspond to 400-600 INTERNIST-1 style manifestations. During the past year, two fellows in Computer Medicine, Drs. Lynn Soffer and Fred Masarie, have converted all INTERNIST-1 findings into the new format required by CPCS.

Dr. Miller has also written, over the past year, a new diagnostic program which uses the information in the new knowledge base as a substrate for making diagnoses in internal medicine. The program's behavior is roughly comparable to that of INTERNIST-1 on similar cases in the limited problem domain currently available for testing. This remains an area of continued research activity.

In addition to the aforementioned work in internal medicine, Drs. Gordon Banks and John Vries have been working on the development of a neurological diagnostic component for CADUCEUS. Dr. Banks has developed a neuroanatomic database which contains spatial descriptors for nearly 1,000 neuroanatomic structures and contains information as to their blood supply and function. This database will allow anatomic localization of neurologic lesions. Some of this work for the peripheral nervous system has been done previously by students in our laboratory. The approach to the central nervous system has been to design a set of "symbolic coordinates". In constructing the neuroanatomic database, the human body, including the nervous system, is conceptually partitioned into a set of cubes (boxes). Attached to each cube LISP atom are lists of all of the anatomic structures that are completely and partially contained within the cube, as well as the blood supply to the region. This structure facilitates rapid retrieval of the location of a given anatomic structure as well as rapid localization of possible areas of involvement when there is evidence of dysfunction of one or more neural systems.

The hierarchical arrangement of the nested cubes ensures rapid convergence during searches, because if the sought object is not found in a parent cube, there is no need to search for it in any of the patient's children cubes. The addition of anatomic reasoning may allow parsimonious explanation of multiple manifestations arising from a single lesion, or allow the program to query the user regarding the presence of manifestations of involvement of areas that might be expected to be affected by whatever clinical state the program has under current consideration.

The neuroanatomic database has been successfully complemented on the VAX 11/780. Efforts are currently underway to implement the system on lower cost AI workstations such as the SUN and the PERQ.

Dr. Vries has continued to work on an image processing system based on "octree" encoding. Sean McLinden has developed an interface to the General Electric 9800 series CT scanner that permits direct input of data from the scanner to the octree system. The octree system output consists of 3 dimensional shaded images of CT objects at 1 mm resolution. Three dimensional images containing 2 million pixels can be scaled, translated, and rotated by the system in 30-60 seconds.

An interface to the neuroanatomic database has also been developed that maps the 27-ary tree representational scheme of the database into an octree representational scheme. This has been used to implement an interactive program that allows a user to generate a three dimensional image of the brain by logically ORing database objects.

A prototype system for the automated diagnosis of CT scans has also been implemented. The system uses the flavors package, and the RUP truth maintenance system to reason about the distribution of CT densities in quadtrees (2 dimensional representations) or octrees (3 dimensional representations). Such a system might ultimately provide CADUCEUS with direct access to the diagnostic information in neuro images.

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 and the pediatrics knowledge base has continued to grow.

----Research in progress

There are five 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.

2. The completion and implementation of the improved diagnostic consulting program, CADUCEUS, which has been designed to overcome certain performance problems identified during the past years of experience with the original INTERNIST-I program.

3. Institution of field trials of CADUCEUS on the clinical services in internal medicine at the Health Center of the University of Pittsburgh.

4. Expansion of the clinical field trials to other university health centers which have expressed interest in working with the system.

5. Adaptation of the diagnostic program and data base of CADUCEUS 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.

D. List of relevant publications

1. First, M.B., Soffer, L.J., Miller, R.A.: *QUICK (Quick Index to Caduceus Knowledge*: Using the Internist-1/Caduceus knowledge base as an an electronic textbook of medicine.) Comput. Biomed. Res. April 1985.

2. Miller, R.A.: *Internist-1/CADUCEUS: Problems Facing Expert Consultant Programs. Methods of Information in Medicine.* Schattauer, Stuttgart - New York, Vol. 23, No. 1, January 1984, pp. 9-14.

3. Miller, R.A.: *A Computer-based Patient Case Simulator. Clinical Research.* 1984, 32:651A. (abstract).

4. Miller, R.A., Schaffer, K.F., Meisel, A.: *Ethical and legal issues related to the use of computer programs in clinical medicine.* Annals of Internal Medicine. 1985, 102:529-536.

5. Myers, J.D.: *Educating future physicians: Something old, Something new.* Ohio State Univ. Proceedings of Symposium, Medical Education in the 21st Century. (in press.)

6. Myers, J.D.: *The process of clinical diagnosis and its adaptation to the computer IN The Logic of Discovery and Diagnosis in Medicine.* University of Pittsburgh Series in the Philosophy and History of Science, Univ. of California Press (in press).

7. Pople, H.E.: *CADUCEUS: An Experimental Expert System for Medical Diagnosis. IN The AI Business.* Edited by Patrick H. Winston and Karen A. Prendergast. 1984, pp. 67-80.

E. Funding support

1. Clinical Decision Systems Research Resource
   Harry E. Pople, Jr., Ph.D.
   Professor of Business
   Jack D. Myers, M.D.
   University Professor (Medicine)
   University of Pittsburgh
   Division of Research Resources
   National Institutes of Health

   5 R24 RR01101-08
   07/01/80 - 06/30/85 - $1,607,717
   07/01/84 - 06/30/85 - $354,211

2. CADUCEUS: A Computer-Based Diagnostic Consultant
   Harry E. Pople, Jr., Ph.D.
   Professor of Business
   Jack D. Myers, M.D.
   University Professor (Medicine)
   University of Pittsburgh
   National Library of Medicine
   National Institutes of Health

   5 R01 LM03710-05
   07/01/80 - 06/30/85 - $817,884
   07/01/84 - 06/30/85 - $210,091

3. Neurologic Consultation Computer Program
   Gordon E. Banks, M.D., Ph.D.
   Assistant Professor of Medicine
   National Library of Medicine - New Investigator
   National Institutes of Health

   5 R23 LM03889-03
   04/01/82 - 03/31/85 - $107,675
   04/01/84 - 03/31/85 - $35,975

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A, B. Medical Collaborations and Program Dissemination Via SUMEX

CADUCEUS remains in a stage of research and 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 CADUCEUS. 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. Profiles of added diseases plus other information is first incorporated into the medical knowledge base at SUMEX before being transferred into our newer information structures for CADUCEUS on the VAX. This sequence retains the operative capability of INTERNIST-1 as a computerized "textbook of medicine" for educational purposes.

B. Justification and Requirements for Continued SUMEX Use

Our use of SUMEX will obviously decline with the installation of our VAX and the use of personal work stations. 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 CADUCEUS is developed here.

Our best prediction is that our project will require continued access to the 2060 for the next two to three years and we consider such access essential to the future development of our knowledge base. After that time, our work can probably be accomplished on our VAX and personal work stations such as Symbolics. The imposition of fees for the use of SUMEX facilities would seem to involve unnecessary book-keeping and probably would detract from the use of SUMEX, which is currently so efficient and pleasant.

Our team hopes to remain as a component of the SUMEX community and to share experiences and developments.

C. Needs and Plans for Other Computing Resources Beyond SUMEX-AIM

Our predictable needs in this area will be met by our dedicated VAX computer and newly acquired personal work stations.

D. Recommendations for Future Community and Resource Development

Whether a program like CADUCEUS, when mature, will be better operated from centralized, larger computers or from the developing self contained personal computers is difficult to predict. For the foreseeable future it would seem that centralized, advanced facilities like SUMEX will be important in further program development and refinement.

6.2.2. CLIPR - Hierarchical Models of Human Cognition

Hierarchical Models of Human Cognition (CLIPR Project)

Walter Kintsch and Peter G. Polson
University of Colorado
Boulder, Colorado

I. SUMMARY OF RESEARCH PROGRAM

A. Project Rationale

The two CLIPR projects have made progress during the last year. The prose comprehension project has completed one major project, and is designing a prose comprehension model that reflects state-of-the-art knowledge from psychology (van Dijk & Kintsch, 1983) and artificial intelligence. During the last three years, Polson, in collaboration with Dr. David Kieras of the University of Michigan, has continued work on a project studying the psychological factors underlying device complexity and the difficulties that nontechnically trained individuals have in learning to use devices like word processors. They have developed formal representations of a user's knowledge of how to operate a device and of the user-device interface (Kieras & Polson, in Press) and have completed several experiments evaluating their theory (Polson & Kieras, 1984, 1985).

B. Technical Goals

The CLIPR project consists of two subprojects. The first, the text comprehension project, is headed by Walter Kintsch and is a continuation of work on understanding of connected discourse that has been underway in Kintsch's laboratory for several years. The second, the device complexity project is headed by Peter Polson in collaboration with David Kieras of the University of Michigan. They are studying the learning and problem solving processes involved in the utilization of devices like word processors or complex computer controlled medical instruments (Kieras & Polson, in Press)

The goal of the prose comprehension project is to develop a computer system capable of the meaningful processing of prose. This work has been generally guided by the prose comprehension model discussed by van Dijk & Kintsch (1983), although our programming efforts have identified necessary clarifications and modifications in that model (Kintsch & Greeno, 1985; Fletcher, 1985; Walker & Kintsch, 1985; Young, 1985). In general, this research has emphasized the importance of knowledge and knowledge-based processes in comprehension. We hope to be able to merge the substantial artificial intelligence research on these systems with psychological interpretations of prose comprehension, resulting in a computational model that is also psychologically respectable.

The goal of the device complexity project is to develop explicit models of the user-device interaction. They model the device as a nested automata and the user as a production system. These models make explicit kinds of knowledge that are required to operate different kinds of devices and the processing loads imposed by different implementations of a device.

C. Medical Relevance and Collaboration

The text comprehension project impacts indirectly on medicine, as the medical profession is no stranger to the problems of the information glut. By adding to the research on how computer systems might understand and summarize texts, and determining ways by which the readability of texts can be improved, medicine can only be helped by research on how people understand prose. Development of a more thorough understanding of the various processes responsible for different types of learning problems in children and the corresponding development of a successful remediation strategy would also be facilitated by an explicit theory of the normal comprehension process.

The device complexity project has two primary goals: the development of a cognitive theory of user-device interaction in including learning and performance models, and the development of a theoretically driven design process that will optimize the relationships between device functionality and ease of learning and other performance factors (Polson & Kieras, 1983, 1984, 1985). The results of this project should be directly relevant to the design of complex, computer controlled medical equipment. They are currently using word processors to study user-device interactions, but principles underlying use of such devices should generalize to medical equipment.

Both the text comprehension project and the device complexity project involve the development of explicit models of complex cognitive processes; cognitive modeling is a stated goal of both SUMEX and research supported by NIMH.

Several other psychologists have either used or shown an interest in using an early version of the prose comprehension model, including Alan Lesgold of SUMEX's SCP project, who is exporting the system to the LRDC Vax. We have also worked with James Greeno -- another member of the SCP project -- on a project that will integrate this model with models of problem solving developed by Greeno and others at the University of California, Berkeley. Needless to say, all of this interaction has been greatly facilitated by the local and network-wide communication systems supported by SUMEX. The mail system, of course, has also enabled us to maintain professional contacts established at conferences and other meetings, and to share and discuss ideas with these contacts.

D. Progress Summary

The version of the prose comprehension model of 1978 (Kintsch & van Dijk, 1978), which originally was realized as a computer simulation by Miller & Kintsch (1980), has been extended in a major simulation program by Young (1985). Unlike the earlier program, Young includes macroprocessing in her model, and thereby greatly extends the usefulness of the program. It is expected that this program will be widely useful in studies of prose where a detailed theoretical analysis is desired.

The general theory has been reformulated and expanded in van Dijk & Kintsch (1983). This research report of book length presents a general framework for a comprehensive theory of discourse processing. It has been applied to an interesting special case, the question of how children understand and solve word arithmetic problems, by Kintsch & Greeno (1985). A simulation for this model, using INTERLISP, has been supplied in Fletcher (1985).

The device complexity project is in its third year. They have developed an explicit model for the knowledge structures involved in the user-device interaction, and they are developing simulation programs. Their preliminary theoretical results are described in Kieras & Polson (in Press). They have also completed several experiments evaluating the theory (Polson & Kieras, 1984, 1985) and have shown that number of productions predicts learning time and that number of cycles and working memory operations predicts execution time for a method.

E. List of Relevant Publications

1. Fletcher, R. C.: *Understanding and solving word arithmetic problems: A computer simulation.* Technical Report NO. 135, Institute of Cognitive Science, Colorado, 1984.

2. Kieras, D.E. and Polson, P.G.: *The formal analysis of user complexity.* Int. J. Man-Machine Studies, In Press.

3. Kintsch, W. and van Dijk, T.A.: *Toward a model of text comprehension and production.* Psychological Rev. 85:363-394, 1978.

4. Kintsch, W. and Greeno, J.G.:*Understanding and solving word arithmetic problems.* Psychological Review, 1985, 92, 109-129.

5. Miller, J.R. and Kintsch, W.: *Readability and recall of short prose passages: A theoretical analysis.* J. Experimental Psychology: Human Learning and Memory 6:335-354, 1980.

6. Polson, P.G. and Kieras, D.E.: *Theoretical foundations of a design process guide for the minimization of user complexity.* Working Paper No. 3, Project on User Complexity, Universities of Arizona and Colorado, June, 1983.

7. Polson, P.G. and Kieras, D.E.: *A formal description of users' knowledge of how to operate a device and user complexity.* Behavior Research Methods, Instrumentation, & Computers, 1984, 16, 249-255.

8. Polson, P.G. and Kieras, D.E.: *A quantitative model of the learning and performance of text editing knowledge.* Proceedings of the CHI 1985 Conference on Human Factors in Computing. San Francisco, April 1985.

9. van Dijk, T.A. and Kintsch, W.:*STRATEGIES OF DISCOURSE COMPREHENSION.* Academic Press, New York, 1983.

10. Young, S.: *A theory and simulation of macrostructure.* Technical Report No. 134, Institute of Cognitive Science, Colorado, 1984.

11. Walker, H.W., Kintsch, W.: *Automatic and strategic aspects of knowledge retrieval.* Cognitive Science, 1985, 9, 261-283.

F. Funding Support Status

1. Text Comprehension and Memory
   Walter Kintsch, Professor, University of Colorado
   National Institute of Mental Health - 5 R0l MH15872-14-16
   7/1/84 - 6/30/87: $145,500 (direct)
   7/1/83 - 6/30/84: $56,501

2. Understanding and solving word arithmetic problems
   Walter Kintsch, Professor, University of Colorado
   National Science Foundation
   8/1/83 - 7/31/86: $200,000

3. The Application of Cognitive Complexity Theory to
   the Design of User Interface Architectures
   David Kieras, Associate Professor, University of Michigan

Peter G. Polson, Professor, University of Colorado
International Business Machines Corporation
1/1/85 - 12/31/85: $250,000 (direct+indirect)

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Sharing and Interactions with Other SUMEX-AIM Projects

Our primary interaction with the SUMEX community has been the work of the prose comprehension group with the AGE and UNITS projects at SUMEX. Feigenbaum and Nii have visited Colorado, and one of us (Miller) attended the AGE workshop at SUMEX. Both of these meetings have been very valuable in increasing our understanding of how our problems might best be solved by the various systems available at SUMEX. We also hope that our experiments with the AGE and UNITS packages have been helpful to the development of those projects.

We should also mention theoretical and experimental insights that we have received from Alan Lesgold and other members of the SUMEX SCP project. The initial comprehension model (Miller & Kintsch, 1980) has been used by Dr. Lesgold and other researchers at the University of Pittsburgh, as well as researchers at Carnegie-Mellon University, the University of Manitoba, Rockefeller University, and the University of Victoria.

B. Critique of Resource Management

The SUMEX-AIM resource is clearly suitable for the current and future needs of our project. We have found the staff of SUMEX to be cooperative and effective in dealing with special requirements and in responding to our questions. The facilities for communication on the ARPANET have also facilitated collaborative work with investigators throughout the country.

III. RESEARCH PLANS

A. Long Range Projects Goals and Plans

The goal of the prose comprehension project is to develop a computer system capable of the meaningful processing of prose. This work has been generally guided by the prose comprehension model discussed by van Dijk & Kintsch (1983), although our programming efforts have identified necessary clarifications and modifications in that model (Kintsch & Greeno, 1985; Fletcher, 1985; Walker & Kintsch, 1985; Young, 1985). In general, this research has emphasized the importance of knowledge and knowledge-based processes in comprehension. We hope to be able to merge the substantial artificial intelligence research on these systems with psychological interpretations of prose comprehension, resulting in a computational model that is also psychologically respectable.

The primary goal of the device complexity project is the development of a theory of the processes and knowledge structures that are involved in the performance of routine cognitive skills making use of devices like word processors. We plan to model the user-device interaction by representing the user's processes and knowledge as a production system and the device as a nested automata. We are also studying the role of mental models in learning how to use them.

B. Justification and Requirements for Continued SUMEX Use

Both the prose comprehension and the user-computer interaction projects have shifted their actual simulation work from SUMEX to systems at the University of Colorado and the University of Michigan. Both projects use Xerox 1108 systems continuing their work in INTERLISP. However, we consider our continued access to SUMEX critical for the successful continuation of these projects.

Access to SUMEX provides us with continued contact with the SUMEX community, which is especially critical for the prose comprehension project. Knowledge representation languages, e.g. UNITS, and other tools developed by SUMEX are critical for this project. Alternative sources of such software are typically unsatisfactory because the systems have only been developed for use on one project and are typically very poorly documented and less than completely debugged. We hope that our continued membership in the community will be offset by the input that we have been and will continue to provide to various projects: our relationship has been symbiotic, and we look forward to its continuation.

Access to SUMEX's mail facilities are critical for the continued success of these projects. These facilities provide us with the means to interact with colleagues at other universities. Kintsch is currently collaborating with James Greeno, who is at the University of California at Berkeley, and Polson's long-term collaborator, David Kieras, is at the University of Michigan. In addition, our access to the Xerox 1108 (Dandelion) user's community is through SUMEX.

We currently use four computing systems for the VAX 11/780, and three Xerox 1108s, one of which is at the University of Michigan. The VAX is used primarily to collect experimental data designed to evaluate the simulation models and to do necessary statistical analysis.

C. Needs and Plans for Other Computational Resources

SUMEX provides us with two critical needs. The first is communication, which we discussed in the preceding paragraph. The second is technical advice and access to various knowledge representation languages like UNITS.

We envisage our future needs to be communication currently served by the SUMEX 2060 and technical advice and necessary software provided by the SUMEX staff.

D. Recommendations for Future Community and Resource Development

Our future needs are for the SUMEX-AIM resource to act as a communications crossroad and to develop software and provide technical support for user community work stations. We have no preferences as to how such services are provided either with a communication server on the network or with the central machine like the current 2060.

We will continue to need access to the SUMEX-AIM 2060 in order to access communication networks and to interact with the SUMEX-AIM staff and community.

If communications and access to the staff are provided through some other mechanism, then we would no longer need access to the 2060.

We would be willing to pay fees for using SUMEX communication resources if required by NIH. However, our willingness is price sensitive. Any charges over $1,000 a year would mean we should communicate with people directly by long-distance telephone.

6.2.3. MENTOR Project

MENTOR Project

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 proposed 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 will employ principles of artificial intelligence to create a more valid system for evaluating therapeutic decision-making.

The work in the MENTOR project is intended to be a collaboration between Dr. Blaschke at Stanford and Dr. Speedie at the University of Maryland. Dr. Speedie provides the expertise in the area of artificial intelligence programming. Dr. Blaschke provides the medical expertise. 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 has focused on exploration of the problem of designing the MENTOR system. Work has begun on constructing a system for monitoring potassium in patients with drug therapy that can adversely affect potassium. Antibiotics, dosing in the presence of renal failure, and digoxin dosing have been identified as additional topics of interest.

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 - December 31, 1988 485,134 Total Direct Costs

Current Period: January 1, 1985 - December 31, 1985 147,170 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 and the University of Maryland School of Pharmacy in exploring computer-based monitoring of drug therapy. SUMEX, through its communications capabilities, facilitates this collaboration of geographically separated project participants by allowing development work on a central machine resource 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 issues related to the formulation of the previously mentioned proposal. We expect interactions with other projects to increase significantly once the groundwork has been laid and issues directly related to AI are being addressed. 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. The only concerns we have relate to the state of the documentation on the system and the response time while using TYMNET from the Baltimore, Maryland area. While most aspects of the system are documented the path to a specific piece of information can be somewhat longer than one might expect. With respect to TYMNET, there are often up to 7 second pauses in the middle of transmissions. This can become quite annoying when trying to work with anything more than small bodies of text.

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.

1985 will be spent on prototype development in four content areas, design and implementation of the basic knowledge representation and reasoning mechanisms and preliminary interfacing to existing patient information systems.

B. Justification and Requirements for Continued SUMEX Use

This project needs continued use of the SUMEX facilities for two reasons. First, it provides access to an environment specifically designed for the development of AI systems. The MENTOR project focuses on the development of such a system for drug monitoring that will explore some neglected aspects of AI in medicine. This environment is necessary for the timely development of a well-designed and efficient MENTOR system. Second, access to SUMEX is necessary to support the collaborative efforts of geographically separated development teams at Stanford and the University of Maryland.

The resources of SUMEX are central to the execution of the MENTOR project. A major component of the proposal was access to SUMEX resources and without it, the chances of funding would have been much less. 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 intend to transport the prototype system to a dedicated hardware system that can fully support the the planned system and which can be integrated into the SUMC Hospital Information System. However, no firm decisions have been made about the requirements for this system since many specification and design decisions remain to be made.

D. Recommendations for Future Community and Resource Development

In the brief 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 nation-wide 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 that Stanford. Specifically, the problem of slow throughput on TYMNET needs to be addressed for those users who do not have authorized access to ARPANET. 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 gradual phase-out of the central SUMEX machine over two or three years to be replaced by an efficient, high-speed communications server.

6.2.4. SOLVER Project

SOLVER: Problem Solving Expertise

Dr. P. E. Johnson
Center for Research in Human Learning
University of Minnesota

Dr. W. B. Thompson
Department of Computer Science
University of Minnesota

I. SUMMARY OF RESEARCH PROGRAM

A. Project Rationale

This project focuses upon the development of strategies for discovering and documenting the knowledge and skill of expert problem solvers. In the last several years, considerable progress has been made in synthesizing the expertise required for solving extremely complex problems. Computer programs exist with competency comparable to human experts in diverse areas ranging from the analysis of mass spectrograms and nuclear magnetic resonance (Dendral) to the diagnosis of certain infectious diseases (Mycin).

Design of an expert system for a particular task domain usually involves the interaction of two distinct groups of individuals, "knowledge engineers," who are primarily concerned with the specification and implementation of formal problem solving techniques, and "experts" (in the relevant problem area) who provide factual and heuristic information of use for the problem solving task under consideration. Typically the knowledge engineer consults with one or more experts and decides on a particular representational structure and inference strategy. Next, "units" of factual information are specified. That is, properties of the problem domain are decomposed into a set of manageable elements suitable for processing by the inference operations. Once this organization has been established, major efforts are required to refine representations and acquire factual knowledge organized in an appropriate form. Substantial research problems exist in developing more effective representations, improving the inference process, and in finding better means of acquiring information from either experts or the problem area itself.

Programs currently exist for empirical investigation of some of these questions for a particular problem domain (e.g. AGE, UNITS, RLL). These tools allow the investigation of alternate organizations, inference strategies, and rule bases in an efficient manner. What is still lacking, however, is a theoretical framework capable of reducing dependence on the expert's intuition or on near exhaustive testing of possible organizations. Despite their successes, there seems to be a consensus that expert systems could be better than they are. Most expert systems embody only the limited amount of expertise that individuals are able to report in a particular, constrained language (e.g. production rules). If current systems are approximately as good as human experts, given that they represent only a portion of what individual human experts know, then improvement in the "knowledge capturing" process should lead to systems with considerably better performance.

In order to obtain a broad view of the nature of human expertise, the SOLVER project includes studies in a variety of complex problem solving domains in addition to medicine. These include law, auditing, business management, plant pathology, and expert system design. We have observed that despite the apparent dissimilarities in these problem solving areas there is reason to believe that there are underlying principles of expertise which apply broadly. Our project seeks to investigate these principles and to create tools to make use of that knowledge in practical expert systems.

B. Medical Relevance and Collaboration

Much of our research has been and will continue to be directly focused on medical AI problems. GALEN, our experimental expert system in pediatric cardiology, is achieving expert levels of performance. Dr. Connelly is initiating a project to develop an expert system based platelet transfusion therapy monitoring program. Dr. Spackman is completing a doctoral thesis on the automated acquisition of rule knowledge in medical microbiology.

Some of our research has focused on problems in diagnostic reasoning and expertise in domains other than medicine. However, our experience indicates that principles of expertise and relevant knowledge engineering tools can cut across task domains. GALEN is demonstrably a useful expert system implementation tool designed in the medical diagnostic task domain. Developments from our work in other domains affecting problems such as automated knowledge acquisition through rule induction and reasoning by analogy will have medical relevance.

Collaboration with Dr. James Moller in the Department of Pediatrics, Dr. Donald Connelly in the Department of Laboratory Medicine, at the University of Minnesota. Dr. Connelly has become a SUMEX user and is teaching a course in medical informatics. He has also initiated a project to create an expert system in platelet transfusion therapy. Collaboration with Dr. Eugene Rich and Dr. Terry Crowson at St. Paul Ramsey Medical Center. Dr. Kent Spackman is a post-doctoral fellow in medical informatics who is completing a Ph.D. thesis in Artificial Intelligence. Dr. Spackman is a resident at the University of Minnesota Hospitals and collaborates with the SOLVER project.

C. Highlights of Research Progress

Accomplishments of This Past Year -- Prior research at Minnesota on expertise in diagnosis of congenital heart disease has resulted in a theory of diagnosis and an embodiment of that theory in the form of a computer simulation model, *Galen*, which diagnoses cases of congenital heart disease [Thompson, Johnson & Moen, 1983]. Continuing development and research with GALEN have led to results in analyzing Garden Path problems in medical diagnosis. Such problems are ones in which an initial solution is later proved to be incorrect. Successful solution of such problems depends upon rejecting an initial incorrect response in favor of a later appropriate one. Errors in Garden Path Problems are generally not due to a lack of knowledge but rather to a confusion over the conditions under which specific rules apply. GALEN was used to identify and test strategies for avoiding Garden Path errors as well as the specific clinical knowledge needed to overcome Garden Path errors in diagnostic reasoning. [Johnson, Moen, and Thompson, 1985].

Galen is descended from two earlier programs written here at Minnesota: *Diagnoser* and *Deducer* [Swanson, 1977]. Deducer is a program that builds hemodynamic models of the circulatory system that describe specific diseases. The models are built by using knowledge about how idealized parts of the circulatory system are causally related. Diagnoser is a recognition-driven program that performs diagnoses by successively hypothesizing one or more of these models and matching them against patient data. The models that match best are used as the final diagnosis. A series of experiments carried out at Minnesota have shown that Diagnoser/Deducer performs as well (and sometimes better) than expert human cardiologists [Johnson et al., 1981].

Despite their early successes, Diagnoser and Deducer did not have a clear, comprehensible structure that is required for the kind of experiments we wish to perform. Galen was built to remedy this problem, taking advantage of the experience gained in the design of Diagnoser and Deducer. Additional discussion of the structure of GALEN can be found in prior annual reports and in the relevant publications.

To determine the generality of our model of expertise in diagnostic reasoning, we are also investigating domains outside medicine. As with our work in congenital heart disease, we have concentrated on the design of mechanisms for structuring problem specific knowledge and for focusing limited computational resources.

One of the Principal Investigators has published results of a study in Expertise in Trial Advocacy, discussing the significance of current research in expertise in legal problem-solving. [Johnson, Johnson, and Little, 1985] Research on legal expertise in corporate acquisition problems has also been investigated. The results of that research suggest that expert corporate acquisition attorneys differ from novices in their greater reliance on internalized norms, prototypes and heuristics. Both expert and novice attorneys in the study went beyond the information provided in task cues in interpreting and predicting actions and situation scripts in the simulated problems. The subjects reasoned heuristically as well as logically. Differences between attorneys in different specialty areas were not large suggesting that the subjects within a domain of problem solving such as legal reasoning acquire meta level reasoning skills that apply to issues within and outside their areas of specialization.

Research is also being completed in a study of cognitive strategies used in making strategic decisions in business. Corporate acquisitions were again used as the context in which to examine expertise. Twenty-four executive subjects were asked to perform an experimental task in which they evaluate companies as candidates for acquisition. The goals of the research are to test for the existence of specialty-related reasoning strategies and to determine the importance of strategic and financial information in problem formulation, problem structuring and choice of strategies in problem solving.

Research in Progress --

Since human experts are notoriously poor at describing their own knowledge, our work requires the creation of problem solving tasks through which experts can reveal criteria for initiating specific hypotheses and methods for investigating those hypotheses.

Current techniques of representing hypotheses and their expectations for diagnosis do not, however, provide much detailed information about the control processes experts use to guide their reasoning. Such control processes typically incorporate highly refined heuristics about which the experts are almost wholly unaware. New research is being proposed to investigate these control structures in legal reasoning, specifically in reasoning by analogy in appellate decision making. Reasoning by analogy appears to be an important inference tool used by experts in many domains as a fundamental problem solving tool. The ability to form plausible analogies lies at the heart of much of the expert ability to be generative when faced with unfamiliar problems. This research will include the implementation of a cognitive simulation of the reasoning by analogy process based upon data obtained by observation of experts solving problems. The results of the simulation will be validated by comparison with human subject data.

We are also investigating several research questions relevant to the architecture of Galen. We have designed an interface to Galen so that users who are unfamiliar with the inner workings of the program can interactively enter case data. Designing the interface raised questions about what forms of data are necessary to adequately and completely represent all possible cases.

One project to test the extensibility of GALEN into other domains is being conducted by a graduate student in the Graduate School of Management. His thesis, Auditing Internal Controls: A computational model of the review process, includes the construction of a working expert system using GALEN. The objective of this study is to formulate and test a model of the processes employed by audit managers and partners in reviewing and evaluating internal accounting controls.

Another project explores the extension of the GALEN architecture into a problem in plant pathology. The main purpose of this research is to find out how the basic postulates about expert reasoning made in Galen hold in a second diagnostic domain. The problem domain chosen for this purpose is Plant Pathology. In collaboration with Professor Paul Teng of the Plant Pathology Department of the University of Minnesota a prototype knowledge base has been implemented. Currently, the knowledge base can diagnose ten potato diseases and has 124 rules. The system is going through evaluation and fine tuning to bring it up to an expert performance level. This system will be useful in the Extension Service at the Plant Pathology department at the University of Minnesota, which provides diagnostic information to farmers over the phone lines.

Dr. Spackman's thesis is entitled "Induction of classification rules under the guidance of comprehensibility-enhancing logical structures and diagnostic performance goals." The purpose of this research is to study and implement methodologies for the automated generation of comprehensible decision rules from empiric data, with emphasis upon logic-based knowledge representation formats and upon problems drawn from the domain of medicine. This work builds upon some of the machine learning methodologies developed at the University of Illinois by R. S. Michalski and others.

This work addresses two shortcomings of previous work on induction of classification rules. These are, first, lack of comprehensibility of the induced rules, and second, lack of flexibility in specifying the diagnostic performance (sensitivity, specificity, or efficiency) desired for the rules that are to be derived.

Comprehensibility of the derived rules or descriptions can be enhanced by imposing restrictions upon the format which the rules may take. For example, the restriction of rules to a unate boolean function format allows the induction of rules that can often be simplified to a "criteria table" type of representation. The type of diagnostic performance a rule must have will depend upon its purpose, and specifying the purpose may allow inductive inference algorithms to trade off small decrements in diagnostic performance for large increments in comprehensibility, or to increase their robustness in the face of noisy or uncertain data.

Successful development of these techniques will lead to enhanced capabilities for deriving rule bases for expert classification systems from empiric data, and will provide new methods for the conceptual analysis of data.

Preliminary results have been obtained for the problem of deriving rules for the identification of bacteria based upon their biochemical profiles in the medical microbiology lab. Other problem domains under investigation are the analysis and interpretation of endocrine laboratory tests, and the induction of rules for the diagnosis of congenital heart disease, for comparison with the rules used in GALEN.

Research is also under way in methods of automating knowledge acquisition in pediatric cardiology. This is being done as thesis research by Paul Krueger. The objective of the research is to design, implement, and test a computerized procedure to derive from examples a nonmonotonic set of rules for an expert classification system. Systems using such rules are generally more efficient than those using monotonic classification processes and more closely approximate psychological models as well.

The research proposes a process for automated learning of preliminary rulebases subject to a set of efficiency constraints which are consistent with a formally defined, psychologically plausible model of classification. The constraints include an upper bound on the amount of information required to explain observations not accounted for by the current set of beliefs, and a lower bound on the degree of inconsistency allowed in the knowledge base at any given time. It will be shown that these constraints can be used to guide the automated determination of both the content and organization of the rules of expert classification systems. The result is behavior that is more focused and efficient, and more closely duplicates the lines of reasoning of domain experts.

A representational formalism for classification knowledge bases based upon a nonmonotonic logic of belief called "autoepistemic logic" (Moore, 1985) is proposed. Having thus defined a representation for the knowledge base the research will propose a methodology for instantiating its concepts within a given application domain. The general approach is to use heuristics to identify from a set of input examples various contextual situations that occur and the types of rules to associate with them. The rule acquisition module (RAM) is then tested in two different application domains. The resulting expert systems will be evaluated for correctness of classification and similarity of their lines of reasoning with those of human experts.

The major conclusion of the research is that constraints similar to those observed in expert human classification processes can be used to guide the empirical induction of efficient expert system rulebases. Supporting this conclusion is the elucidation of a formal nonmonotonic model of classification, and the design and subsequent testing of the Rule Acquisition Module and expert systems derived by it.

D. List of Relevant Publications

1. Connelly, D. and Johnson, P.E.: *Medical problem solving.* Human Pathology, 11(5):412-419, 1980.

2. Elstein, A., Gorry, A., Johnson, P. and Kassirer, J.: *Proposed Research Efforts.* IN D.C. Connelly, E. Benson and D. Burke (Eds.), CLINICAL DECISION MAKING AND LABORATORY USE. University of Minnesota Press, 1982, pp. 327-334.

3. Feltovich, P.J.: *Knowledge based components of expertise in medical diagnosis.* Learning Research and Development Center Technical Report PDS-2, University of Pittsburgh, September, 1981.

4. Feltovich, P.J., Johnson, P.E., Moller, J.H. and Swanson, D.B.: *The Role and Development of Medical Knowledge in Diagnostic Expertise.* IN W. Clancey and E.H. Shortliffe (Eds.), READINGS IN MEDICAL AI, Addison-Wesley, 1984, pp. 275-319.

5. Johnson, P.E.: *Problem Solving.* IN ENCYCLOPEDIA OF SCIENCE AND TECHNOLOGY, McGraw-Hill (in press).

6. Johnson, P.E., Moen, J.B., and Thompson, W.B.: *Garden Path Errors in Medical Diagnosis.* IN Bloc, L. and Coombs, M.J. (Eds.), COMPUTER EXPERT SYSTEMS, Springer-Verlag (in press).

7. Johnson, P.E.: *Cognitive Models of Medical Problem Solvers.* IN D.C. Connelly, E. Benson, D. Burke (Eds.), CLINICAL DECISION MAKING AND LABORATORY USE. University of Minnesota Press, 1982, pp. 39-51.

8. Johnson, P.E.: *What kind of expert should a system be?* J. Medicine and Philosophy, 8:77-97, 1983.

9. Johnson, P.E., *The Expert Mind: A new Challenge for the Information*

Scientist IN Th. M. A. Bemelmans (Ed.), INFORMATION SYSTEM DEVELOPMENT FOR ORGANIZATIONAL EFFECTIVENESS, Elsevier Science Publishers B. V. (North-Holland), 1984.

10. Johnson, P.E., Severance, D.G. and Feltovich, P.J.: *Design of decision support systems in medicine: Rationale and principles from the analysis of physician expertise.* Proc. Twelfth Hawaii International Conference on System Science, Western Periodicals Co. 3:105-118, 1979.

11. Johnson, P.E., Duran, A., Hassebrock, F., Moller, J., Prietula, M., Feltovich, P. and Swanson, D.: *Expertise and error in diagnostic reasoning.* Cognitive Science 5:235-283, 1981.

12. Johnson, P.E. and Hassebrock, F.: *Validating Computer Simulation Models of Expert Reasoning.* IN R. Trappl (Ed.), CYBERNETICS AND SYSTEMS RESEARCH. North-Holland Publishing Co., 1982.

13. Johnson, P.E. and Thompson, W.B.: *Strolling down the garden path: Detection and recovery from error in expert problem solving.* Proc. Seventh IJCAI, Vancouver, B.C., August, 1981, pp. 214-217.

14. Johnson, P.E., Hassebrock, F. and Moller, J.H.: *Multimethod study of clinical judgement.* Organizational Behavior and Human Performance 30:201-230, 1982.

15. Moller, J.H., Bass, G.M., Jr. and Johnson, P.E.: *New techniques in the construction of patient management problems.* Medical Education 15:150-153, 1981.

16. Sedlmeyer, R.L., Thompson, W.B. and Johnson, P.E.: *Knowledge-based fault localization in debugging.* The Journal of Systems and Software, vol. 3, no. 4 (Dec 83) pp. 301-307, Elsevier.

17. Sedlmeyer, R.L., Thompson, W.B. and Johnson, P.E.: *Diagnostic reasoning in software fault localization.* Proc. Eighth IJCAI, Karlsruhe, West Germany, August, 1983.

18. Smith, K.A., Farm, B., Johnson, P.E.: *Surface: A prototype expert system for selecting surface analysis techniques.* Proceedings of IEEE Conference on Computers and Comm., 1985.

19. Swanson, D.B.: *Computer simulation of expert problem solving in medical diagnosis.* Unpublished Ph.D. dissertation, University of Minnesota, 1978.

20. Swanson, D.B., Feltovich, P.J. and Johnson, P.E.: *Psychological Analysis of Physician Expertise: Implications for The Design of Decision Support Systems.* In D.B. Shires and H. Wold (Eds.), MEDINFO77, North-Holland Publishing Co., Amsterdam, 1977, pp. 161-164.

21. Thompson, W.B., Johnson, P.E. and Moen, J.B.: *Recognition-based diagnostic reasoning.* Proc. Eighth IJCAI, Karlsruhe, West Germany, August, 1983.

E. Funding and Support

Work on the SOLVER project is currently supported by a grant from the Control Data Corporation to Paul Johnson ($90,000; 1983-85) and by a grant from the Microelectronics and Information Sciences Center at the University of Minnesota to Paul Johnson, William Thompson, James Slagle (Dept. of Computer Science), Harry

Wechsler (Electrical Engineering), and Albert Yonas (Institute for Child Development) ($500,000; 1984-85).

Research in medical informatics is supported, in part, by a training grant from the National Library of Medicine, LM-00160, in the amount of $712,573 for the period 1984-1989. Dr. Connelly and Prof. Johnson are participants in this grant. The post doctoral fellowship of Dr. Spackman is funded by this grant.

"Expert system techniques for analyzing and evaluating internal accounting controls." McKnight Foundation, $13,000 (1984-5). Paul E. Johnson and Andrew D. Bailey.

Dwan Family Fund, University of Minnesota Medical School, $6,000 (1985) to Paul Johnson for research assistant funding on the GALEN project.

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Medical Collaborations and Program Dissemination via SUMEX

Work in medical diagnosis is carried out with the cooperation of faculty and students in the University of Minnesota Medical School and St. Paul Ramsey Medical Center.

B. Sharing and Interactions with Other SUMEX-AIM Projects

William Clancey, Stanford University, acted as a reviewer of the MEIS Intelligent Systems Project in September, 1984 at the University of Minnesota. The Principal Investigators in the SOLVER project are also principal investigators in that project.

Paul Johnson was a panel member at the SUMEX-AIM conference in Columbus, Ohio in 1984. Dr. Connelly and two graduate students associated with the SOLVER PROJECT also attended the conference.

III. RESEARCH PLANS

A. Project Goals and Plans

Near term -- Our research objectives in the near term can be divided in three parts. First, we are committed to the design, implementation, and evaluation of Galen, as described above. We have completed an interactive front end so that physicians can directly enter patient data, and Galen's knowledge base is currently being "tuned" with the help of Dr. James Moller, an expert physician collaborator from the University of Minnesota Pediatric Cardiology Clinic, the Diagnoser program, and with expert physicians. We believe that GALEN has passed through phases of expertise assessment and cognitive simulation and that it is now approaching a level of performance that will qualify it as a true expert system. An objective now is to extend the explanation capability of GALEN. We are initiating a new investigation into two aspects of expert problem solving that relate to the interaction between a problem solving system and its environment: *"query generation"* and *explanation*. Some simple expert systems proceed from a fixed set of input data to an evaluation of that data. For most problem domains, however, the space of possibly relevant information is large, and some or all of this information may have costs associated with its acquisition. Thus, computational and other costs can be reduced by some mechanism which intelligently selects appropriate queries designed to solicit information that is relevant and cost effective in terms of the problem being solved. Expert systems for complex problem domains must also be able to generate explanations for their actions. Unless the system operates in an entirely autonomous manner, users must be apprised of the rationale for system actions. There is a particular need for explanations tailored for system users rather than system designers.

Experienced experts are typically quite proficient at asking relevant questions, even when the criteria for relevance is difficult to specify. These experts use heuristics capable of keying on selected aspects of data already examined and on the current problem state in order to select the next needed query. We propose to incorporate these heuristics into a *"query generation knowledge base"* . This knowledge base can be thought of as a form of domain specific meta-knowledge. It contains rules by which the problem state can be efficiently evaluated in order to determine the next course of action. By basing these rules on actual expert knowledge and experience, it will often be possible to bypass the combinatorial complexity associated with either blind search or optimization techniques.

Our approach to explanation starts from the premise that substantially different forms of explanation are required within a single expert system. The type of explanation is distinguished both by the level of sophistication of the person receiving the explanation and by whether that person is principally interested in the specific problem being solved or in the internal working of the expert system. Less sophisticated users of the system are likely to have only a superficial understanding of the nature of the system being diagnosed and will require explanations in terms of simplified system properties with which they are familiar. Expert users will require information about significant details of the state of the system being diagnosed and the causal relationships that connect system state with observable symptoms. Designers and maintainers of the expert system require explanations in terms of the actual lines of reasoning used to arrive at a decision.

We will be focusing principally on providing explanations for system *users* rather than system *designers*. Explanations for users must be phrased in terms of the system being diagnosed. Descriptions of the system itself are more important that descriptions of the reasoning strategies used to understand the system. For example, many diagnostic tasks are efficiently approached utilizing recognition-based reasoning strategies using knowledge arising from empirical association. Experts (or possibly automatic learning systems) learn to associate particular interpretations with particular patterns in the data. For many problem domains, knowledge of this sort is quite powerful, providing accuracy without the complexity associated with causal reasoning. The user of such a system, however, requires explanations in terms of causality. This suggest a two-step process. Problem solving is done using a recognition-based strategy. Explanations are generated by combining the results of this process with additional, causally-based explanation knowledge.

Our second objective consists of making extensions to the knowledge capturing strategies developed in our original work in medical diagnosis. In the near term this work will examine descriptive strategies in which experts attempt to use a formalized language to express what they know (e.g. production rules), observational strategies in which experts perform tasks designed to reveal information from which a theory of task specific expertise can be built, and intuitive strategies in which either experts behave as knowledge engineers or knowledge engineers attempt to perform as pseudo experts. The research projects of Dr. Spackman and Paul Krueger which have been discussed previously are both directed toward this objective.

Our third near term objective will be to investigate one of the central problems of recognition based problem solving, how to classify problems when solving them. Questions related to problem classification which we will be examining include: What patterns do experts and novices detect in a problem that allows them to classify it as an instance of a problem type that is already known? How does an expert make an initial choice of the level of abstraction to be used in solving a problem? How can an expert recover from an initial incorrect choice of levels? How can the difference between causal and prototypic modes of reasoning be modeled as differences in levels of abstraction, and how can a common model for these two types of reasoning be constructed? We will be pursuing these questions in the areas of problem solving like law, auditing, and management, as well as in medicine.

Long range -- Our long range objective is to improve the methodology of the "knowledge capturing" process that occurs in the early stages of the development of expert systems when problem decomposition and solution strategies are being specified. Several related questions of interest include: What are the performance consequences of different approaches, how can these consequences be evaluated, and what tools can assist in making the best choice? How can organizations be determined which not only perform well, but are structured so as to facilitate knowledge acquisition from human experts? In the coming year we will be exploring these questions in areas of design and management as well as in law, management and medicine.

B. Justification and Requirements for Continued SUMEX Use

Our current model development takes advantage of the sophisticated Lisp programming environment on SUMEX. Although much current work with Galen is done using a version running on a local VAX 11/780, we continue to benefit from the interaction with other researchers facilitated by the SUMEX system. We expect to use SUMEX to allow other groups access to the Galen program. We also plan to continue use of the knowledge engineering tools available on SUMEX.

We are working toward a Commonlisp implementation of the GALEN system and expect to rely heavily on Commonlisp for future projects.

One of our students implemented a demonstration legal expert system in EMYCIN using the SUMEX resource, and we still find that the resource is valuable for making available major systems which we do not have locally, such as EMYCIN.

C. Needs and Plans for Other Computing Resources Beyond SUMEX-AIM

Our current grant from MEIS has permitted us to purchase four Perq 2 AI workstations for our Artificial Intelligence laboratory. The availability of Commonlisp on these machines is one reason why we expect to make use of that language in the future.

SUMEX will continue to be used for collaborative activities and for program development requiring tools not available locally.

D. Recommendations for Future Community and Resource Development

As a remote site, we particularly appreciate the communications that the SUMEX facility provides our researchers with other members of the community. We, too, are moving toward a workstation based development environment, but we hope that SUMEX will continue to serve as a focal point for the medical AI community. In addition to communication and sharing of programs, we are interested in development of Commonlisp based knowledge engineering tools. The continued existence of the SUMEX resource is very important to us.

6.3. Stanford Pilot 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.

6.3.1. CAMDA Project

CAMDA Project

CAMDA Research Staff:

Prof. Samuel Holtzman, Co-PI	**Engineering-Economic Systems**
Prof. Ronald A. Howard, Co-PI	**Engineering-Economic Systems**
Prof. Ross Shachter	**Engineering-Economic Systems**
Leonard Bertrand	**Engineering-Economic Systems**
Jack Breese	**Engineering-Economic Systems**
Kazuo Ezawa	**Engineering-Economic Systems**
Keh-Shiou Leu	**Engineering-Economic Systems**
Seok Hui Ng	**Engineering-Economic Systems**
Emilio Navarro	**Engineering-Economic Systems**
Dr. Adam Seiver	**Engineering-Economic Systems**
Joseph Tatman	**Engineering-Economic Systems**
Dr. Emmet Lamb	**School of Medicine**
Dr. Robert Kessler	**School of Medicine**
Dr. Frank Polansky	**School of Medicine**

Associated faculty:

Prof. Edison Tse	**Engineering-Economic Systems**

I. SUMMARY OF RESEARCH PROGRAM

A. Project Rationale

The Computer-Aided Medical Decision Analysis (CAMDA) project is an attempt to develop intelligent medical decision systems by combining the descriptive generality of expert-system technology with the normative power of decision analysis.

B. Medical Relevance and Collaboration

The primary effort of the CAMDA project during 1984 and early 1985 has been focused on the design and implementation of RACHEL, an intelligent decision system for infertile couples. This system is designed to help patients and physicians deal with difficult medical treatment choices. RACHEL is being developed in close cooperation with the Engineering-Economic Systems Department, the Obstetrics and Gynecology Department, and the Surgery Department (Urology Division), all at Stanford.

In addition to the development of RACHEL, there are several active research programs within the CAMDA project. One such program is aimed at developing a representation for dynamic decision processes (such as those faced by cancer patients) that do not necessarily satisfy the Markov assumption. Another is concentrating on the development of fast algorithms for the solution of general decision problems.

A recent addition to our research project is a program to design cost-effective strategies for monitoring the recurrence of bladder cancer.

C. Highlights of Research Progress

C.1 Accomplishments this past year

We have successfully implemented a pilot-level version of RACHEL. As we define it, a pilot system is one where the essential algorithms work individually as well as interactively with one another, operating with knowledge that is representative of the system's domain. Such a system lacks two important elements that must exist within a prototype-level implementation: an extensive knowledge base, and a front end usable by trained users who may not be familiar with the details of the system.

As part of the development of RACHEL, we have developed a facility to construct individualized models of the patient's preferences over the set of possible outcomes of an infertility therapy. This facility operates in two consecutive stages. The first stage constructs a parametric model from a library of plausible model elements. A typical consideration at this stage is whether to explicitly account for the patient's lifetime. For instance, a treatment strategy which involves surgery would warrant such explicit consideration, whereas a therapy consisting strictly of drugs would not. The second stage in the preference model development process involves the assessment of specific parametric values. These values are obtained directly from the patient to ensure that the overall preference model genuinely reflects his or her desires.

It is important to note that since the preference model is built to fit the specific needs of each case, the interaction between the patient and the system is short and well-focused. In particular, the patient is only asked to respond to a few (about five to ten) questions. These questions are selected so that their relevance to the case is intuitively obvious from the patient's point of view.

Also as a part of RACHEL, we have developed a knowledge base dealing with the decisions faced by the subset of infertile couples whose inability to conceive has been traced to a blockage of the Fallopian tubes of the female partner. In particular, the knowledge in RACHEL deals with the choice between two important procedures pertinent to this condition: laparotomy and in-vitro fertilization.

Another accomplishment during this past research year has been the improvement of our influence-diagram solution procedure. In its original form, this procedure essentially took a brute-force approach to the solution of well-formed influence diagrams. Although its solutions were mathematically correct, the program was inefficient in terms of both computational time and storage requirements. In its current implementation, the program is considerably more efficient and has an adequate front end which makes it accessible to a fairly wide class of users. Empirical results indicate that the size and complexity of problems that can be represented and solved with the system not only exceed the bounds of its original design, but are comparable and possibly superior to those of the best commercially available decision-analytic software.

Similarly, RACHEL's inference engine has been improved in several important ways. Prominent among these are a means for attaching general procedures at any point in the inference process, a variety of built-in procedures for the acquisition and display of information coupled with a facility for controlling these procedures (i.e., for the control of ASKability and TELLability), and a simple explanation mechanism.

C.2 Research in progress

The RACHEL system continues to be developed along four distinct directions: the efficiency and flexibility of RACHEL's inference engine are being improved, its explanation mechanism is being enhanced, RACHEL's facility for the development of patient preference models is being upgraded, and its knowledge base is being enlarged.

As it is currently implemented, the inference engine used by RACHEL is quite inefficient. This inefficiency is, to some extent, a deliberate design choice since the engine was designed to be very general and highly modular. Thus, there are many procedural redundancies and much unnecessary baggage in the programs that implement it. Now that we have a clearer idea of how the engine is to be used we have redesigned it by doing away with some of the original generality and modularity in favor of a more efficient process. Furthermore, the new design emphasizes and enhances particularly useful engine features such as its ASKability and its TELLability.

A further enhancement to RACHEL's inference engine concentrates on the system's ability to explain its line of reasoning. The original design only responds to online "why" queries by displaying its dynamic goal stack. In its new form, the engine allows offline as well as online queries in both "why" and "how" formats.

Beyond traditional explanation capabilities, we are exploring possible means to explain decision-theoretic inferences. In particular, we are trying to understand how to explain decision recommendations that are based on the maximization of expected utility to users unfamiliar with decision theory. Our current research indicates that a promising way to do this is to break down large decision problems into smaller, more manageable pieces whose formal solution can be checked against intuition. Although still at an early stage, this line of research seems to be on the path of eliminating an important barrier to the widespread use of normative decision techniques.

An exciting area of current interest is the improvement of RACHEL's facility for the creation and assessment of parametric models of patient preferences. In particular, we are trying to increase the generality of RACHEL's model library to account for acute as well as chronic conditions and to simplify the corresponding assessment process. This simplification is based on the notion that a better understanding of the major concerns of patients can help us redesign the questions asked by RACHEL so that they are closer to the specific experiences of individual patients. As part of this effort, we expect to have significant contact with actual patients to ensure the clinical relevance of our research.

A fourth area where RACHEL is being enhanced is the expansion of its medical and decision-analytic knowledge bases. Planned additions include further knowledge about the treatment of tubal blockage (including more data on in-vitro fertilization procedures and an ability to consider a wider class of patients) and a new packet of knowledge dealing with deterministic sensitivity analysis.

In addition to the development of RACHEL, there are several active research programs within the CAMDA project. One such program is aimed at developing a representation for dynamic decision processes (such as those faced by cancer patients) that do not necessarily satisfy the Markov assumption. This research has led to a generalization of influence diagrams which allows multiple value nodes. This generalization makes it possible for complex sequential decision processes (whose solution would otherwise be infeasible) to be efficiently solved.

Another research program within the CAMDA project is the development of fast algorithms for the solution of decision problems formulated as influence diagrams. In general, the solution of an influence diagram (i.e., the calculation of a recommended decision strategy) is obtained by the repeated application of an operation, known as "removal", to all nodes in the diagram other than the value node. The removal of a node in the diagram is a generalization of the foldback operation needed to solve a decision tree. With rare exceptions, the order in which nodes are removed from a diagram is not unique. Current results indicate that significant reductions in the computational burden of solution can be achieved by controlling the order in which diagram nodes are selected for removal.

At a more fundamental level, we are exploring the consolidation of the predicate calculus with probabilistic logic. Of particular interest is the design of an integrated inference engine that performs logical inferences within a probabilistic framework. A central problem in this research is the definition of universal and existential quantification in probabilistic terms.

A recent addition to our research project is a program to design cost-effective strategies for monitoring the recurrence of bladder cancer. We expect this research to interact with our ongoing search for more effective models of patient preferences.

D. Publications

1. Holtzman, S.:*A Model of the Decision Analysis Process*, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1981.

2. Holtzman, S.:*A Decision Aid for Patients with End-Stage Renal Disease*, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1983.

3. Holtzman, S.:*On the Use of Formal Models in Decision Making*, Proc. TIMS/ORSA Joint Nat. Mtg., San Francisco, May, 1984.

4. (*) Holtzman, S.: *Intelligent Decision Systems*, Ph.D. Dissertation, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1985.

5. Shachter, R.: *Evaluating Influence Diagrams*, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1984.

6. Shachter, R.: *Automating Probabilistic Inference*, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1984.

E. Funding Support

E.I Principal Funding Source

E.I.1. Title of gift

"Research on Intelligent Decision Systems".

E.I.2. Principal investigator

Samuel Holtzman, Ph.D.
Consulting Assistant Professor
Department of Engineering-Economic Systems
Stanford University

E.I.3. Funding source

Olivetti Advanced Technology Center, Inc.

E.I.5. Funding amount

$33,400 (Direct Costs), unrestricted.

E.II Additional Funding Source

E.I.1. Title of gift

"Cost-effective strategies in monitoring for recurrence of bladder cancer"

E.II.2. Principal Investigators

Ross Shachter, Ph.D. -- PI
Assistant Professor
Department of Engineering-Economic Systems
Stanford University

Linda Shortliffe, M.D. -- Co-PI
Palo Alto Veterans Administration Hospital

Dan Kent, M.D. -- Co-PI
Division of General Internal Medicine
Stanford University Medical Center

Samuel Holtzman, Ph.D. -- PI: CAMDA Project (SUMEX)
Consulting Assistant Professor
Department of Engineering-Economic Systems
Stanford University

E.II.3. Funding agency

Stanford's American Cancer Society Institutional Research Grant Committee

E.II.5. Total award

$4634 (Direct Costs), for the year starting April 1, 1985

E.III Other Funding

E.III.2 Donated Equipment

The CAMDA project has access to the facilities of the Decision Systems Laboratory (DSL) in the Department of Engineering-Economic Systems, and constitutes the laboratory's most active research project. The DSL maintains several terminals, printers and personal computers for research on the development of computer-based decision systems. The majority of the terminals and printers were donated to the DSL by Qume Corporation. Olivetti Advanced Technology Center, Inc., has made four M24 personal computers and two high-quality printers available to the DSL on a "Beta-test-site" basis. MAD Computer, Inc., has also contributed to the support of the CAMDA project through the consignment of a MAD-1 personal computer.

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

II.A Medical Collaborations and Program Dissemination Via SUMEX

Since its inception, the CAMDA project has benefited from an active relationship among decision analysts, computer scientists, and members of the Stanford medical community. In particular, RACHEL is being developed in close cooperation with physicians in the Infertility Clinic at Stanford. Other programs within the CAMDA project such as our research on the form and use of medical preference models are being done in cooperation with physicians at the Palo Alto Veterans Administration Hospital and at El Camino Hospital.

II.B. Sharing and Interactions with other SUMEX-AIM Projects

II.B.1 SUMEX-AIM 1984 Workshop:

Samuel Holtzman participated in the 1984 AIM workshop in Columbus, Ohio. In addition to the presentation of a summary of CAMDA research, he had many opportunities to interact with workshop participants on an informal basis. Of particular interest were several discussions with members of the MIT/TUFTS group interested in medical decision analysis which have led to an interchange of ideas that continues to this date.

II.B.2 Decision Systems Laboratory Research Meetings

As part of the CAMDA project, we have instituted a weekly research meeting for those interested in the design and implementation of computer-based decision systems. This weekly meeting has become a very active forum for the presentation of research results. The following topics of direct relevance to medical decision making were presented during the last two academic quarters.

Date	Speaker	Topic
03-OCT-84	Ross Shachter	Probabilistic Inference
17-OCT-84	Jack Breese	Dempster-Shafer Theory
24-OCT-84	Kazuo Ezawa	Efficiency in Solving Influence Diagrams
07-NOV-84	Majid Khorram	Fuzzy Sets and Decision Making
14-NOV-84	Dan Kent	Utility Theory Underlying Physicians' Treatment Thresholds: HELP!
21-NOV-84	Yann Bonduelle	Explanation in Decision Systems
09-JAN-85	Ross Shachter	What Do You Call the Offspring of SUPERID and INFLUENCE?
23-JAN-85	Doug Logan	The Value of Probability Assessment
06-FEB-85	Seok Hui Ng	Minimal Tumor Follow-up Examination Schedule for Recurrent Bladder Cancer Patients.
13-FEB-85	Keh-Shiou Leu	TEREISIAS' Explanation Facility
06-MAR-85	Joe Tatman	Algorithm for Decision Processes Optimization
13-MAR-85	Gerald Liu (UC)	Knowledge Structure in Evidential Reasoning

II.B.3 Course in Medical Decision Analysis

A new course in medical decision analysis, taught by Prof. Samuel Holtzman, is being offered for the first time during the Spring quarter of 1985. The course is offered jointly by the Engineering-Economic Systems Department, the Medical Information Sciences Program, and the Computer Science Department. The objective of the course is to expose students to the practice of decision analysis for clinical purposes and to introduce them to the design and use of computer-based medical decision tools.

II.C. Critique of Resource Management

The CAMDA project is heavily dependent upon the availability of the SUMEX computing resource. The physical facility as well as the staff of SUMEX-AIM are excellent. In particular, it has been a pleasure to deal with Ed Pattermann, who is invariably courteous, responsive to our needs, and effective in his actions. We will certainly miss him now that he has moved to industry. Pam Ryalls has also provided much needed help in managing the CAMDA project in a manner that is friendly and efficient.

As an update to last year's report, the previously reported Ethernet deficiencies have been corrected. This improvement was part of a campus-wide effort to improve Stanford's computer network which directly affected our campus connection to SUMEX. The system load on SUMEX continues to be heavy, although it appears to be somewhat lower than it was last year. The ability of the CAMDA project to use the DECSYSTEM-2020 machine operated by SUMEX (referred to as TINY) has had a significant effect on our ability to demonstrate our systems during normal business hours, further reducing our frustration with the main system's load.

III. RESEARCH PLANS

III.A Project Goals and Plans

During the upcoming year, we intend to enhance four specific elements of the RACHEL system: its inference mechanism, its explanation facility, its ability to model patient preferences, and its medical and decision-analytic knowledge bases. Furthermore, we intend to continue to improve our understanding of normative decision methodologies, with particular emphasis on the use of these methodologies for computer-based decision support. Section I.C.2 describes the near-term goals of the CAMDA project in more detail. Our long-term goal remains that of designing and implementing usable, fully-validated and documented systems for medical decision support.

III.B Justification and Requirements for Continued SUMEX Use

The CAMDA project is truly interdisciplinary. It draws on elements of decision analysis, artificial intelligence, and medical science. The project has the potential to contribute to each of these disciplines in important ways.

In particular, the CAMDA project is likely to lead to the development of tools and techniques that greatly improve the quality of decision making in medicine. For instance, RACHEL explicitly considers uncertainty, decision alternatives, and patient preferences in developing recommendations. In spite of its generality, RACHEL's interaction with the user is sufficiently terse and simple to support the claim that systems based on its methodology can be effective clinical decision tools. Much of the simplicity and terseness of RACHEL's operation is a direct consequence of the AI foundations of the system's design.

The heavy reliance of the CAMDA effort on artificial intelligence technology make SUMEX-AIM an ideal environment in which to pursue this research.

III.C Needs and Plans for other Computing Resources beyond SUMEX-AIM

The CAMDA project has access to four Olivetti M24 and one MAD-1 personal computers (IBM-PC type) as well as to one Apple Macintosh (128K) computer. In addition, we continue to search for funds to acquire one or more state-of-the-art LISP machines.

III.D Recommendations for Future Community and Resource Development

What would be the effect of imposing fees for using SUMEX resources (computing and communications) if NIH were to require this?

A major benefit provided by the existing SUMEX-AIM facility is the availability of very low-cost computing resources. Access to these resources is granted primarily on the basis of an assessment of the value of the proposed research to the overall goal of making artificial intelligence a useful medical tool. Imposing fees for using SUMEX would prevent users with modest means from obtaining access to the facility on the basis of merit alone.

Do you have plans to move your work to another machine workstation and if so, when and to what kind of system?

The CAMDA project has access to several personal computers for its research. These machines include Olivetti M24's (marketed as the A.T.&T. personal computer in the U.S.) and a MAD-1 personal computer -- all of which are compatible with the IBM-PC. In addition, the project has purchased an Apple Macintosh. These machines are used as a supplement to the SUMEX mainframe, and are not intended to replace it.

6.3.2. REFEREE Project

REFEREE Project

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

Byron W. Brown, Ph.D.
Dept. of Biostatistics
Stanford University

Daniel E. Feldman, Ph.D., M.D.
Department of Medicine
Stanford University

I. SUMMARY OF RESEARCH PROGRAM

A. Project Rationale

The goal of this project is two-fold: (a) use existing AI methods to implement an expert system that can critique medical journal articles on clinical trials, and (b) in the long term, develop new AI methods that extract new medical knowledge from the clinical trials literature. In order to accomplish (a) we are building the system in three stages.

1. System I will assist in the evaluation of the quality of a single clinical trial. The user will be imagined to be the editor of a journal reviewing a manuscript for publication, but the program will be tested on a variety of readers, including clinicians, medical scientists, medical and graduate students, and clerical help.

2. System II will assist in the evaluation of the effectiveness of the treatment or intervention examined in a single published clinical trial. The user will be imagined to be a clinician interested in judging the efficacy of the treatment being tested in the trial.

3. System III will assist in the evaluation of the effectiveness of a single treatment examined in a number of published clinical trials.

B. Medical Relevance

The burden of "keeping up with the literature" is particularly onerous in the practice of medicine and in medical research [62, 63]. Reading the abstracts in a few journals and selecting several key articles for a rapid survey are the best that most clinicians can hope to accomplish each week. The time and effort necessary for a thorough and critical reading of even a few research reports are not available.[1] Sackett reports that to keep up with the 10 leading journals in internal medicine a clinician must read 200 articles and 70 editorials per month [63]. It was also estimated that the biomedical

---

[1] In an informal check on this intuition two of us, with considerable training in analyzing clinical trials (BWB and DEF) timed critical readings of a five page article on a clinical trial in the New England Journal of Medicine [4]. Our times were 30 and 120 minutes.

literature is expanding at a compound rate of 6% to 7% per year, or doubling every 10 - 15 years [63, 59]. Furthermore, even if more time were available the statistical and epidemiological skills necessary for critical reading are not part of most clinicians' repertoires[1]; and yet decisions about which therapy to use, what intervention to adopt, or what advice to give patients must be based on a combination of clinical experience and published literature. But the existing literature is often confusing and contradictory [42]and publication in the most prestigious medical journals does not guarantee freedom from serious methodologic flaws and erroneous conclusions [44, 18]. Any assistance to the clinician must deal with both the problem of the vastness of the literature and the quality of the research report. Similar problems are faced by the editors of medical journals, swamped with manuscripts to review and evaluate, and by research scientists and academicians trying to stay abreast of the developments in their fields. How can they cover more and yet evaluate better and more consistently? Clearly any machine assistance would be welcome.

C. Highlights of Progress

This project is just getting started.

Preliminary work has been done on REFEREE [23], a prototype expert system for determining the quality of a clinical trial report, and the efficacy of the intervention evaluated in the trial. REFEREE is written in EMYCIN, a rule-based programming language which allows rapid prototyping of a consultation system that gives advice to a user. It presupposes that a knowledge base about the problem area has been constructed, which usually involves codifying an expert's knowledge.

The basic format of a REFEREE session is fairly simple. The reader is asked a series of questions pertaining to the paper and the study described. The answers given are used to rate the overall quality of the paper and the probable efficacy of the treatment described. (See sample dialogs below).

In the first version of REFEREE, after the program has finished with its chain of questions and deductions, the quality of the paper and the efficacy of the drug are given to the user as a "merit score", an integer between 0 and 10, with 10 indicating the highest quality. Additionally, the user is provided with a series of English language messages indicating the main flaws detected in the paper. The merit score was used because the expert system makes its judgements by using a weighted average of values assigned to each aspect of the paper being critiqued. As the user answers the consultant's questions, the answers are given individual merit scores. For example, if the user's answer indicate that experimental blinding was done correctly, the paper is given a high score in the blinding category. When all merit score assignments have been made, the total merit score is calculated as a weighted average of the categorical merit scores, with those categories that are more crucial to a good paper or clinical trial being given a higher weight.

The final result of this calculation is a number between 1 and 10 which serves as a quality measure for the paper or the treatment. A 1 indicates low quality; a 10 indicates the highest quality. An integer as a final result, however, can be very cryptic. It is usually quite difficult, given just an integer, to understand or believe the findings of the consultant. It was discovered quite early that users, when presented with just the bare merit score of the paper, would want to know why the paper was rated in the way it was. For this reason, English language statements are given to the user, indicating the nature of the main flaws of the paper. In each category, if the calculated merit score is

[1]A recent survey of the statistical methods used by authors in the New England Journal of Medicine indicated that 42 per cent of the articles surveyed relied on statistical analysis beyond descriptive statistics [15].

found to be less than an arbitrary minimum, this is noted in a sentence or two, and given to the user at the end of the consultation. In this way, the user not only gets an overall picture of the quality of the paper, but also an indication of the general areas in which the paper was found to be lacking.

Several problems were found in the original version of REFEREE. It was discovered that the use of a weighted average precluded the use of EMYCIN's certainty factors. Because of this, the user would often be forced to choose from a fairly limited set of possible answers to the consultant's questions. The lack of versatility implied by this constraint dictated that a new approach which could make full use of EMYCIN's certainty factors should be used.

In order to do this, the old rule base was scrapped, and a new one was written. Instead of deciding on a rating between one and ten to indicate quality, the new version simply decides whether or not the paper in question is of "high academic and scholarly quality", with an EMYCIN certainty factor modifying the conclusion. For example, in the case of a mediocre paper, the program would conclude that the paper was of "high quality", but only with a certainty of say, .5, on a scale between -1 and 1. Though the words "certainty factor" are used for historical reasons, our final number is the equivalent of a merit score.

While at first glance the two approaches seem similar, the second approach was found to be much more flexible and satisfying from the user's standpoint. Since the conclusion is in terms of the programs certainty that the paper's quality is good, the user may incorporate his or her own uncertainty into the dialogue with the program. This was accomplished by asking mainly yes/no questions, and at all times allowing the user to indicate his or her certainty in the answers given. Thus, if the program asks the user if the quality of the paper's literature review was high, he or she can answer simply "yes" or "no", indicating complete confidence in the answers, or modify a yes/no answer with a certainty factor, indicating that he or she is not completely certain. The user's answers, along with the uncertainty indicated by him or her, will be combined by EMYCIN to give a final conclusion on the paper's quality.

As an example, one of the old-style rules might have been something like this: If the user indicates that the literature review is of "poor quality", conclude that the merit of the paper is 3 with a (built-in) weight of 2. After all the merit values had been calculated, a weighted average, (using built-in weights) would be taken to come to the final merit score. In contrast, one of the new rules would be of the form: If the user gives a "yes" answer to the question "Is the literature review thorough and balanced?", conclude that the paper is of good quality with a certainty of .3. While in the first case the user was limited to a set of possible answers (e.g. excellent, good, poor), the second rule gives the user the opportunity to answer either yes or no, and qualify that answer with any degree of certainty desired. If, in the second rule, the user gives a certainty of less than 1 that the literature review was of good quality, the inferred conclusion about the quality of the paper will be automatically downgraded as well. In other words, if the user expresses uncertainty, the conclusion about the quality of the paper will be less certain.

The new approach, in addition to supplying the user with the ability to express varying degrees of uncertainty, also allows for a hierarchical question structure. At any point, if the user is unclear of the appropriate response, the program can prompt with further, more detailed questions, until a conclusion about the original question can be provided. Conversely, whenever a user is willing to give an answer, the program will refrain from dwelling on the issue and omit its long series of sub-questions. In this manner the amount of detail provided can be individualized.

This current version of REFEREE has two hundred rules and has been tested by the present research team on several papers. It is this program that will be expanded as described in Section III-A. Part of a sample consultation is shown below.

```
   --------MEDICINE-1--------
   The first paper of MEDICINE-1 will be referred to as:

      --------PAPER-1--------

         --------STATISTICS-1--------
         1) What is the size of the control sample?
         ** 25
         2) How many of the subjects in the control sample responded to
            treatment?
         ** 14
         3) What is the size of the test sample?
         ** 23
         4) How many of the subjects in the test sample responded to
            treatment?
         ** 23
...

         --------PLANNING-1--------
         9) Was there an explicit stopping rule defined before the experiment
            was run?
         ** N

         --------RANDOMIZATION-1--------
      10) Was there any mention of the use of randomization in patient
          assignment?
      ** Y
      11) Was the assignment of subjects in the experiment performed blindly?
      ** UNK
...
         --------BLINDING-1--------
      16) Was the experiment double blinded, or was any mention made of
          blinding in the experiment?
      ** Y
      17) Was there any mention of an effort to make the placebo and
          medication as similar as possible?
      ** N
...
The strength of the evidence indicating the efficacy of PAPER-1 is as
     follows:
        There is some evidence for efficacy, but further study is needed.

The general quality of the paper is as follows:
        The current paper is of poor quality.

The flaws of the current paper are as follows:
        A stopping rule was not defined or was not adhered to in the
            experiment.
        The measures taken to evaluate subject compliance were inadequate or
            non-existent.
        Subjects were not randomly assigned treatment groups, seriously
            weakening the validity of the conclusions.
        Though an effort was made to blind the experiment, the techniques
            used were not effective.

The final calculated efficacy of the drug as indicated by the given clinical
     trial (between 0 and 10, with a score of 10 being the highest) is as
     follows:
        5.

The final merit of the current paper is as follows:
        3.

   23) Are there any other papers on MEDICINE-1?
   ** N
   24) Do you want the results of this consultation output to a file?
   ** N
```

E. Funding Support

Grant applications submitted to the NLM:

Title: Understanding and Critiquing Clinical Trials Literature

PI's: Bruce G. Buchanan, Byron W. Brown

Agency: National Library of Medicine (Pending)

Total Amount: $178,923.

Dates: July 1, 1985 - June 30, 1988

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Medical Collaborations

Dr. D. Feldman is a physician and epidemiologist at the Stanford Center for Disease Prevention. Prof. B. Brown is currently teaching a Medical School class on reading medical journal articles.

B. Interactions with other SUMEX-AIM projects

Our interactions have all been through the Knowledge Systems Laboratory where we have discussed design and implementation issues.

C. Critique of Resource Management

The SUMEX staff has been most cooperative in helping get this project started. We have tried to place few demands on the SUMEX staff, but have received prompt answers to all questions.

III. RESEARCH PLANS

A. Goals & Plans

It is proposed to construct three computer-based expert systems to assist a variety of different readers in the evaluation of an extensive but well defined area of the medical literature, clinical trials. It is further proposed to test the hypothesis that such programs will enable a variety of users to read the literature on clinical trials more more critically and more rapidly.

The expert systems will be developed using the EMYCIN programming environment and the production rule approach followed successfully in previous expert systems [24, 36, 43, 48, 6].

The three programs to be developed are separate, but closely related:

1. System I will assist in the evaluation of the quality of a single clinical trial. The user will be imagined to be the editor of a journal reviewing a manuscript for publication, but the program will be tested on a variety of readers, including clinicians, medical scientists, medical and graduate students, and clerical help.

2. System II will assist in the evaluation of the effectiveness of the treatment or intervention examined in a single published clinical trial. The user will

be imagined to be a clinician interested in judging the efficacy of the treatment being tested in the trial.

3. System III will assist in the evaluation of the effectiveness of a single treatment examined in a number of published clinical trials.

Within the duration of this research it is also proposed to test the first two systems against unassisted evaluations by the various categories of readers. The testing will include a formal testing of the programs by comparing the speed and number of flaws found in using the program with similar measurements on unassisted reading. In addition there will be a more informal evaluation by questionnaire of the subjective impressions of users of the program, ascertaining the likelihood of routine use and the value of such a program to the user.

This proposal with its concentration on clinical trials is regarded as the initial step in a more general research goal - building computer systems to help the clinician and medical scientist read the medical literature more critically.

B. Justification for continued SUMEX use

We will continue to use SUMEX for developing the AI methods. We need EMYCIN at the moment because it provides a good environment for building a rule-based system that may grow to many hundreds of rules. EMYCIN is not available on other machines without substantial cost.

C. Need for other computing resources

In the short term we will not need additional resources. Should we decide to implement a new system in a framework other than EMYCIN, we might seek funding to buy a LISP workstation.

D. Recommendations

Although our use has been small, we find the load average on SUMEX often precludes running test cases during the day. We have no specific recommendation, but would like to have access to small amounts of high quality computer time.

6.4. National AIM Pilot Projects

Following is a description of the informal pilot projects currently using the national AIM portion of the SUMEX-AIM resource, pending funding, full review, and authorization.

6.4.1. 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 80 common benign and malignant diseases of the lymph node based on 150 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.

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 10 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 Accomplishments This Past Year

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 graduate students, David Heckerman and Eric Horvitz, designed and implemented the program.

C.1 The PATHFINDER knowledge base

The basic building block of the PATHFINDER knowledge base is the disease profile or *frame*. The disease frame consists of *features* useful for diagnosis of lymph node diseases. Currently these features include histopathological 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*.

Evoking strengths and frequencies are associated with each feature-value pair in a disease profile. We are experimenting with different scales for scoring each feature-value pair, and several methods for combining the scores to form a differential diagnosis. A disease-independent import is also assigned to each feature-value but only a two-valued scale is used. This is because, in PATHFINDER, imports are only used to make boolean or yes/no decisions (see below). In addition to import, PATHFINDER utilizes the concept of *classic* features for a disease -- within each disease frame, the pathologist marks those feature-value pairs which are considered to be part of the classic pattern of the disease.

The PATHFINDER knowledge base contains information about obvious association between features. This information is of the form: "Don't ask about feature x unless feature y has certain values." For example, it wouldn't make sense to ask about the degree or range of follicularity if there are no follicles in the tissue section. The feature links also serve to identify interdependencies among features. Feature interdependence is a problem because it can lead to inaccuracies in scoring hypotheses.

The prototype knowledge base was constructed by Dr. Nathwani. During the beginning 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.

D. Publications Since January 1984

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.

E. Funding Support

Research Grant submitted to National Institutes of Health, March, 1984.
Grant Title:"Computer-aided Diagnosis of Malignant Lymph Node Diseases"
Principal Investigator: Bharat Nathwani

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 TYMNET 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 expert, and then will be extensively tested. A logical next step would be to extend the program to clinical settings, as well as possible extensions of the knowledge base.

Other possible extensions include: developing techniques for simplifying the acquisition and verification of knowledge from experts, 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 hope to enhance the INTERNIST-1 model 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 the use of the program by remote users, and for project coordination. 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 HP equipment is currently unable to allow remote access, and thus the program will have to be maintained on the 2060 for use by all 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 HP9836 workstation. We will need additional file space on the 2060 as we quadruple the size of our knowledge base. 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 fcrth 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 ($5000 or less), we will continue to need a machine like SUMEX to provide others with a chance to experiment with our software.

6.4.2. RXDX Project

RXDX Project

Robert Lindsay, Ph.D.
Michael Feinberg, M.D., Ph.D.
Manfred Kochen, Ph.D.
University of Michigan
Ann Arbor, Michigan

I. SUMMARY OF RESEARCH PROGRAM

A. Project Rationale

We are developing a prototype expert system that could act as a consultant in the diagnosis and management of depression. Health professionals will interact with the program as they might with a human consultant, describing the patient, receiving advice, and asking the consultant about the rationale for each recommendation. The program uses a knowledge base constructed by encoding the clinical expertise of a skilled psychiatrist in a set of rules and other knowledge structures. It will use this knowledge base to decide on the most likely diagnosis (endogenous or nonendogenous depression), assess the need for hospitalization, and recommend specific somatic treatments when this is indicated (e.g., tricyclic antidepressants). The treatment recommendation will take into account the patient's diagnosis, age, concurrent illnesses, and concurrent treatments (drug interactions).

B. Medical Relevance and Collaboration

There has been a growing emphasis in American psychiatry on careful diagnosis using clearly defined clinical criteria (Feighner, et al., 1972; Spitzer, et al., 1975, 1980; Feinberg and Carroll, 1982, 1983). These efforts have led to several sets of criteria for the diagnosis of psychiatric disorders. The "St. Louis" criteria (Feighner, et al., 1972) were succeeded by the Research Diagnostic Criteria (RDC), formulated by researchers from St. Louis and New York (Spitzer, et al., 1975). The RDC led directly to the criteria that are now quasi-official in American psychiatry, DSM-III (Spitzer, et al., 1980). All of these criteria lists were based on a combination of clinical opinion and literature review, and use a decision-tree approach to making a diagnosis. These diagnostic systems have been shown to be acceptably reliable, but their validity remains untested. Other groups have used a multivariate statistical approach to diagnosis. Roth and his colleagues (Carney, et al., 1965) published a discriminant index for distinguishing "endogenous" from "neurotic" depressed patients. This work was repeated by Kiloh, et al. (1972) with much the same results, confirming the findings of Carney, et al. (1965).

We have done similar work, deriving two discriminant indices for separating endogenous depressed patients (unipolar or bipolar) from nonendogenous (neurotic) patients. We cross-validated these indices in separate groups of patients, and also validated them against an external standard, the dexamethasone suppression test (Feinberg and Carroll, 1982, 1983). At the same time, we and others have been further developing this and other biological measures that may differentiate between patients with endogenous and nonendogenous depression. These include neuroendocrine tests such as the dexamethasone suppression test (DST) and quantitative studies of sleep using EEG. Carroll, et al. (1981) have shown that the DST is abnormal in about 67%

of patients with endogenous depression (melancholia) and only 5-10% with nonendogenous (neurotic) depression. Kupfer, et al. (1978) and Feinberg, et al. (1982) have similar results with EEG studies of sleep. These biological markers may be useful for routine clinical use, and can certainly be used as external validating criteria to test the performance of different clinical diagnostic methods, including those mentioned above. Furthermore, we have developed biological criteria for "definitely endogenous" depression and "definitely nonendogenous" depression based on DST and sleep EEG. (Carroll, et al., 1980). Our goal is to use these criteria as an external validating criterion for assessing the performance of various new or different diagnostic schemes, in particular an expert system of the sort we are developing.

C. Highlights of Research Progress

We examined two other SUMEX-based psychiatry projects, the BLUEBOX project of Mulsant and Servan-Schreiber (1984), and the HEADMED project of Heiser and Brooks (1978, 1980). Mulsant and Servan-Schreiber visited us at Michigan and discussed the rationale and progress of their project. Heiser also visited with us and agreed to collaborate with our project as a consultant.

At Michigan, we encoded the Hamilton Rating Scale (Hamilton, 1967) into EMYCIN rules. This is the standard scale (in English) for rating the severity of depression, and many of the items in it are relevant to our consultant program. We moved our work to the AGE system, breaking the Hamilton scale into its component subscales and adding other components to determine patient demographic information, personal and family psychiatric history, and other rating scale information. We then introduced other knowledge sources to construct a differential diagnosis list for psychiatric illnesses based on our expert's taxonomy and methods. We are now focussing on rules that discriminate endogenous from non-endogenous depression. Concurrently we are developing a treatment knowledge base on a LISP workstation. Thus far, the treatment knowledge base contains information about drug therapies, including types, dosages, activities, interactions, and side effects.

We have conducted interviews with patients recently admitted to the University of Michigan Adult Psychiatric Hospital. They are interviewed by Feinberg and the interviews are observed by Lindsay plus a group of psychiatric residents, psychiatrists and psychologists. After the interview, Feinberg is debriefed by Lindsay, and then the others discuss the case. These data are the initial source of the expert knowledge base for our consultant.

D. List of Relevant Publications

This project has not yet produced any publications. The following list contains the references cited above, including our previous publications relevant to the RxDx Project.

1. Carney, M. W. P., Roth, M. and Garside, R. F.:*The diagnosis of depressive syndromes and the prediction of ECT response*, Brit. J. Psychiatry, 111, 659-674, 1965.

2. Carroll, B. J., Feinberg, M., Greden, J. F., Haskett, R. F., James, N. McI., Steiner, M., and Tarika, J.: *Diagnosis of endogenous depression: Comparison of clinical, research, and neuroendocrine criteria*, J. Affect Dis., 2, 177-194, 1980.

3. Carroll, B. J., Feinberg, M., Greden, J. F., Tarika, J., Albala, A. A., Haskett, R. F., James, N. McI., Kronfol, Z., Lohr, N., Steiner, M., de Vigne, J-P, and Young, E.:*A specific laboratory test for the diagnosis of melancholia*, Standardization, validation, and clinical utility. Arch. Gen. Psychiatry, 38, 15-22, 1981.

4. Feighner, J. P., Robins, E., Guze, S. B., Woodruff, R. A., Winokur, G., and Munoz, R.: *Diagnostic criteria for use in psychiatric research*, Arch. Gen. Psychiatry, 26, 57-63, 1972.

5. Feinberg, M. and Lindsay, R. K.: *Expert systems*. Proceedings of the NCDEU Annual Meeting, Key Biscayne, Florida, May 1985.

6. Feinberg, M. and Carroll, B. J.: *Separation of subtypes of depression using discriminant analysis: I. Separation of unipolar endogenous depression from non-endogenous depression*, Brit. J. Psychiatry, 140, 384-391, 1982.

7. Feinberg, M. and Carroll, B. J.:*Separation of subtypes of depression using discriminant analysis. II. Separation of bipolar endogenous depression from nonendogenous ("neurotic") depression*, J. Affective Disorders, 5, 129-139, 1983.

8. Feinberg, M. and Carroll, B.J.: *Biological markers for endogenous depression in series and parallel*, Biological Psychiatry 19:3-11, 1984.

9. Feinberg, M. and Carroll, B.J.: *Biological and nonbiological depression*, Presented at Annual Meeting of the Society of Biological Psychiatry, Los Angeles, May, 1984, Abstract #81.

10. Feinberg, M., Gillin, J. C., Carroll, B. J., Greden, J. F., and Zis, A. P.:*EEG studies of sleep in the diagnosis of depression*, Biological Psychiatry, 17, 305-316, 1982.

11. Heiser, J. F. and Brooks, R. E.:*Design considerations for a clinical psychopharmacology advisor*, Proc. Second Annual Symp. on Computer Applications in Medical Care. New York: IEEE, 1978, 278-285.

12. Heiser, J. F. and Brooks, R. E.:*Some experience with transferring the MYCIN system to a new domain*, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-2, No. 5, 477-478, 1980.

13. Kiloh, L. G., Andrews, G., and Neilson, M.:*The relationship of the syndromes called endogenous and neurotic depression*, Brit. J. Psychiatry, 121, 183-196, 1972.

14. Kupfer, D. J., Foster, F. G., Coble, P., McPartland, R. J., and Ulrich, R. F.:*The application of EEG sleep for the differential diagnosis of affective disorders*, Am. J. Psychiatry, 135, 69-74, 1978.

15. Mulsant, B. and Servan-Schreiber, D.:*Knowledge engineering: A daily activity on a hospital ward*, Computers in Biomedical Research, 1984.

16. Spitzer, R. L., Endicott, J. and Robins, E.: *Research diagnostic criteria*, (2d ed.) New York State Department of Mental Hygiene, New York Psychiatric Institute, Biometrics Research Division, 1975.

17. Spitzer, R. L.: (Ed.)*Diagnostic and statistical manual of mental disorders*, (3d ed.). Washington, D. C.: American Psychiatric Association, 1980.

18. Van Melle, W.:*The EMYCIN Manual*, Computer Science Department, Stanford University, Report HPP-81-16, 1981.

E. Funding Support

We have received support from the Vice-President for Research at the University of Michigan, and from the NIH "Small Grants" Program (Grant Number ro3MH40239-01; Total Direct Costs = $13,850). These funds have enabled us to gather the pilot data for a grant application to be submitted to NIH on July 1, 1985.

II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A. Medical Collaboration and Program Dissemination via SUMEX

We have established via SUMEX a community of researchers who are interested in AI applications in psychiatry. We also have used the message system to communicate with other AI scientists at SUMEX and elsewhere.

B. Sharing and Collaboration with other SUMEX-AIM Projects

Our use of EMYCIN and AGE has been of major importance. In addition, we have worked with Dr. Larry Fagan to learn about his Pathfinder program. We used that program, on SUMEX, to obtain some information for the RxDx project by applying it to data we previously collected on depression symptom frequencies.

C. Critique of Resource Management

We have been using EMYCIN and AGE in our work, and have found these programs very valuable, saving us many hours of programming in LISP. There are some problems with them, many of which center around discrepancies between the versions described in the manuals and the versions actually running on SUMEX. We would suggest that software be more strongly supported than is now the case, if it and SUMEX are to be even more useful to beginners in AI in Medicine.

SUMEX itself has been invaluable. We don't have ready access to any other machine of equal computing power which also has a strongly supported LISP available. Specifically, the LISP compiler available on the Amdahl 5860 here differs from those used at major AI centers such as Stanford and MIT. We have also made good use of the ARPANET connections that SUMEX offers. Feinberg spent a month of his sabbatical working with Prof. Peter Szolovits at MIT, learning about AI in Medicine. This visit was arranged using computer mail through SUMEX. Lindsay and Feinberg were able to continue their collaborative work while the latter was in Cambridge, using the same medium. The alternative would have been days lost in the mails and many dollars spent on phone calls. We have also been able to get help with problems that arise with EMYCIN and AGE using computer mail.

Most of the limitations of SUMEX, and they are often severe, derive from the necessity to access it via TYMNET. Response time is often impossibly slow, and even at its best the delays are annoying and frustrating, even for editing and debugging. For example, editing is limited to a primitive line editor, since EMACS interacts with the network XON/XOFF handshaking in a disastrous way. The staff has not been helpful in solving these network related problems, probably because they do not have to live with them in their own interactions with the system. In any case, many of the problems are beyond the reach of the Sumex staff. The future of long-haul network collaborations depends critically on increased bandwidth and faster response times.

It would have been helpful to us to obtain the AGE system that runs on a Xerox 1108. However, the $530 price, though perhaps modest in comparison to its development costs, was beyond the reach of our budget. It would be helpful if distribution costs for software could be held under $100.

III. RESEARCH PLAN

A. Project Goals and Plans

Our immediate objective is to develop an expert system that can differentiate patients with the various subtypes of depressive disorder, and prescribe appropriate treatment. This system should perform at about the level of a board-certified psychiatrist, i.e. better than an average resident but not as well as a human expert in depression. Eventually, we plan to enlarge the knowledge base so that the expert system can diagnose and prescribe for a wider range of psychiatric patients, particularly those with illnesses that are likely to respond to psychopharmacological agents. We will design the system so that it could be used by non-medical clinicians or by non-psychiatrist MD's as an adjunct to consultation with a human expert. We plan also to focus on problems of the user interface and the integration of this system with other databases.

B. Justification and Requirements for continued SUMEX use

The access to SUMEX resources is essentially our sole means of maintaining contact with the community of researchers working on applications of AI in medicine. Although we plan to move our system to local workstations as soon as we are able, the communications capability of SUMEX will continue to be important.

We anticipate that our requirements for computing time and file space will continue at about the same level for the next year.

C. Needs and Plans for Other Computing Resources

As our project evolves and we run into the limitations of the time-shared SUMEX facility, we anticipate employing different expert systems software. At this time, we are not at a stage to say exactly what that will be, but our project is not sufficiently large that we will be able to mount such a software development project ourselves, so we will depend on development and support elsewhere. Ultimately, when our consultant is made available for field trials and clinical use, it will need to be transported to a personal computer that is large enough to support the system yet inexpensive enough to be widely available. A LISP machine is an obvious candidate. While current prices of the necessary hardware are too high, computer prices are continuing to drop. Our design strategy is to avoid limiting ourselves and our aspirations to that which is affordable today; instead we will attempt to project the growth of our project and the price-performance curve of computing such that they meet at some reasonable point in the future.

D. Recommendations for Future Community and Resource Development

Valuable as the present SUMEX facilities are to us, they are in many ways limited and awkward to use. The major limitation we feel is the difficulty and sometimes the impossibility of making contact with everyone who could be of value to us. We hope that greater emphasis will be put on internetwork gateways. It is important not only to establish more of these, but to develop consistent and convenient standards for electronic mail, electronic file transfers, graphic information transfer, national archives and data bases, and personal filing and retrieval (categorization) systems. The present state of the art feels quite limiting, now that the basic concepts of computer networking have become available and have proved their potential.

We expect that the role of the SUMEX-AIM resource will continue to evolve in the direction of increased importance of communication, including graphical information, electronic dissemination of preprints, and database and program access. The need for computer cycles on a large mainframe will diminish. We hope to have continued access to the system for communication, but do not anticipate continued use of it as a LISP computation server beyond the next year or eighteen months.

If fees for using SUMEX resources were imposed, this would have a drastically limiting effect on the value of the system to us. Even if we had a budget to purchase such services, the inhibiting effect of having a meter running would cause us to make less use of it that we should. We have been conscious of the costs of the system and feel that we have not used it imprudently, even though we have not directly borne its costs.

Appendix A

Stanford Knowledge Systems Laboratory

ARTIFICIAL INTELLIGENCE RESEARCH IN THE KNOWLEDGE SYSTEMS LABORATORY (Incorporating the Heuristic Programming Project)

Stanford University
Department of Computer Science/Department of Medicine
April 1985

The Knowledge Systems Laboratory (KSL) is an artificial intelligence research laboratory of about 90 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 increasing complexity and diversity of the research now under way. The KSL is a modular laboratory, consisting of five collaborating yet distinct groups with different research themes:

- **The Heuristic Programming Project (HPP)**, Professor Edward A. Feigenbaum, scientific director -- blackboard systems, concurrent system architectures for AI, and the modeling of discovery processes. Executive director: Robert Engelmore. Research scientists: Harold Brown, Byron Davies, Bruce Delagi, Peter Friedland, Barbara Hayes-Roth, and H. Penny Nii. Consulting professor: Richard Gabriel.
- **The HELIX Group**, Professor Bruce G. Buchanan, scientific director -- machine learning, transfer of expertise, and problem solving. Faculty: Paul S. Rosenbloom (joint appointment, Computer Science and Psychology). Research scientists: James Brinkley, William J. Clancey, Barbara Hayes-Roth.
- **The Medical Computer Science (MCS) Group**, Professor Edward H. Shortliffe, scientific director (Department of Medicine with courtesy appointment in Computer Science) -- research on and advanced application of AI to medical problems; includes the Medical Information Sciences (MIS) program. Research scientist: Lawrence M. Fagan.
- **The Logic Group**, Professor Michael R. Genesereth, scientific director -- formal reasoning and introspective systems. Research scientist: Matthew L. Ginsberg.
- **The Symbolic Systems Resources Group (SSRG)**, Thomas C. Rindfleisch, scientific director (joint appointment, Computer Science and Medicine) -- research on and operation of computing resources for AI research, including the SUMEX facility. Assistant director: William J. Yeager.

Tom Rindfleisch serves as KSL project 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?
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?
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? How can problem-solving systems be combined with pedagogical and user knowledge to implement intelligent tutoring systems?
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?

Research and Academic Programs

CURRENT RESEARCH PROJECTS

The following list of projects now under way within the five 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
415-497-3444

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), Harold Brown, Byron Davies (TI), Bruce Delagi (DEC), Richard Gabriel, Penny Nii, Sayuri Nishimura, Jim Rice, Eric Schoen, Jerry Yan.
- **Knowledge-Based VLSI Design Project** -- Study the hierarchical design process involved in the development of complex very large scale integrated circuits.
  Personnel: Harold Brown (contact), Jerry Yan.
- **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).
- **MOLGEN** -- Study the processes of scientific theory formation and modification, using recently developed models of genetic regulation as an example.
  Personnel: Peter Friedland (contact), Charles Yanofsky (Biological Science), Peter Karp.

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 (Nuclear Magnetic Laboratory), Jim Brinkley, Barbara Hayes-Roth, Russ Altman, Olivier Lichtarge.
- **NEOMYCIN/GUIDON2** -- Develop knowledge representation and explanation capabilities for the computer-aided teaching of diagnostic reasoning.
  Personnel: Bill Clancey (contact), Stephen Barnhouse, Diane Hasling, David C. Wilkins.
- **SOAR** -- Develop a general production-system-based problem-solving architecture that integrates reasoning, domain expertise, learning, and planning of problem-solving strategies.
  Personnel: Paul Rosenbloom (contact), Andrew Golding, Amy Unruh.
- **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), Li-Min Fu, Russell Greiner, Ramsey Haddad, David C. Wilkins.

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, Gregory Cooper, Jay Ferguson, Christopher Lane, Janice Rohn, Homer Chin, Holly Jimison, Curt Langlotz, Mark Musen, Glenn Rennels.

- PATHFINDER -- Develop a knowledge-based system for diagnosis of lymph node pathology.
  Personnel: Ted Shortliffe, Bharat Nathwani (USC), Larry Fagan (contact), David Heckerman, Eric Horvitz.

The Logic Group

- **Metalevel Representation System (MRS)** -- Study logic-based introspective programs that can reason about and control their own problem-solving activities.
  Personnel: Mike Genesereth (contact), Matt Ginsberg, Russ Greiner, Ben Grosof, Yung-Jen Hsu, David E. Smith, Devika Subramanian, Richard Treitel.
- **The DART/HELIOS Project** -- Study an integrated design environment that includes capabilities for design specification, refinement, and validation; fabrication engineering; and failure diagnosis and testing.
  Personnel: Mike Genesereth (contact), Glenn Kramer (Fairchild), Narinder Singh.
- **Intelligent Agent Project** -- Study planning and problem-solving activities for an intelligent interface between human users and complex computing environments.
  Personnel: Mike Genesereth (contact), Matt Ginsberg, Jeff Finger, Jeff Rosenschein, Jock Mackinlay, Vineet Singh.
- **Intelligent Task Automation** -- Build a program that can use the description of a manufacturing task to develop a plan by which a robot can carry out the task.
  Personnel: Mike Genesereth (contact), Matt Ginsberg, Jeff Finger, David E. Smith, Richard Treitel.

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), Bill Croft, Frank Gilmurray, Christopher Schmidt, Andrew Sweer, Israel Torres, Bob Tucker, Nicholas Veizades, Bill Yeager.
- **Financial Resource Management** -- Develop an expert system for financial resource planning.
  Personnel: Tom Rindfleisch (contact), Bruce Buchanan.

Other Projects

The KSL also has close ties to collaborative projects. These include PIXIE, developing an intelligent tutoring system, under Derek Sleeman in the School of Education, and RADIX, studying discovery of knowledge from databases, under Bob Blum in Computer Science.

STUDENTS AND SPECIAL DEGREE PROGRAMS

Graduate students are an essential part of the research productivity of the KSL. Currently 41 students are working with our projects centered in Computer Science and another 12 students are working with the MCS/MIS programs in Medicine. Of the 41 working in Computer Science, 25 are working toward Ph.D. degrees, and 16 are working toward M.S. degrees. A number of students are pursuing interdisciplinary programs and come from the Departments of Engineering, Mathematics, Education, and Medicine.

Because of the highly interdisciplinary and experimental nature of KSL research, two special degree programs have been established:

The **Medical Information Sciences (MIS)** program is 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.

The **Master of Science in Computer Science: Artificial Intelligence (MS:AI)** program is 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 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 students, 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 (MYCIN), cancer chemotherapy management (ONCOCIN), pulmonary function evaluation (PUFF), machine fault diagnosis (DART), VLSI design (KBVLSI/PALLADIO), and molecular biology (MOLGEN). Some of these programs rival human experts in solving problems in restricted domains. A number of projects have developed generalized software tools for representing and using knowledge; of these, EMYCIN, AGE, MRS, and BB1 are available to outside research groups. Some of our systems and tools (e.g., DENDRAL, PUFF, UNITS, and EMYCIN) are now also being adapted for commercial development and use in the burgeoning 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 major managerial posts and conference chairmanships in both the American Association for Artificial Intelligence (AAAI) and the International Joint Conference on Artificial Intelligence (IJCAI).

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 received the IJCAI Computers and Thought award, and in 1978, Ed Feigenbaum received the National Computer Conference Most Outstanding Technical Contribution award. In 1981, Ted Shortliffe's book *Computer-Based Medical Consultation: MYCIN* was identified as the most frequently cited work in the IJCAI-81 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, Randy Davis and Doug Lenat were named among the 100 most promising U.S. scientists under 40 by a prestigious scientific panel assembled by Science Digest. Also 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 Information.

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 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 host machines include a DEC 2060 and 2020 running TOPS-20 (these are the core of the national SUMEX biomedical computing resource) and a VAX 11/780 running UNIX. Our growing complement of Lisp machines includes more than 25 Xerox 1100's, a Xerox Dorado, a Symbolics LM-2, eight Symbolics 3600's, and five Hewlett-Packard 9836's. Network printing, file, gateway, and terminal interface services are provided by dedicated machines ranging from VAX 11/750's to 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 Tymnet. Funding for these resources comes principally from DARPA and NIH.

Appendix B

Resource Operations and Usage Data

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

1. Overall resource loading data (page 294).
2. Relative system loading by community (page 295).
3. Individual project and community usage (page 298).
4. Network usage data (page 305).
5. System reliability data (page 305).

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 severe changes in most of the graphs. This is due to many reasons which I will mention briefly here ;

1. Even though the TENEX operating system used on the KI-10 was a forerunner of the current Tops20 operating system, the Tops20 system is still different from TENEX is many ways. Tops20 uses a radically different job scheduling mechanism, different methods for computing monitor statistics, different I/O routines, etc. In general, it can not be assumed that statistics measured on the TENEX system correlate one to one with similar statistics under Tops20.
2. The KL-10 processor on the 2060 is a faster processor than the KI-10 processor used previously. Hence, a job running on the KL-10 will use less CPU time than the same job running on the KI-10. This aspect is further complicated by the fact that the SUMEX KI-10 system was a dual processor system.
3. The SUMEX-AIM Community was changing during the time of the transfer to the 2060. The usage of the GENET community on SUMEX had just been phased out. This part of the community accounted for much of the CPU time used by the AIM community. Since the purchase of the 2060 was partially funded by the Heuristic Programming Project (HPP), an additional number of HPP Core Research Projects started using the 2060, increasing the Stanford communities usage of the machine. And finally, the move to the 2060 occurred during a pivotal time in the community when more and more projects were either moving to their own local timesharing machines, or onto specialized Lisp workstations. It also was the time for the closure of many long time SUMEX-AIM projects, like DENDRAL and PUFF/VM.

Any conclusions reached by comparing the data before and after February, 1983 should be done with caution. The data is included in this years annual report mostly for casual comparison.

Also, it should be noted that monthly statistics are not available for this past year because of problems with the accounting program at this writing. The appropriate average data quantity for the year is shown instead for each month so the graphs appear to be "flat" in the area corresponding to the current period.

Overall Resource Loading Data

The following plot displays total CPU time delivered per month. This data includes usage of the KI-TENEX system and the current DECsystem 2060.

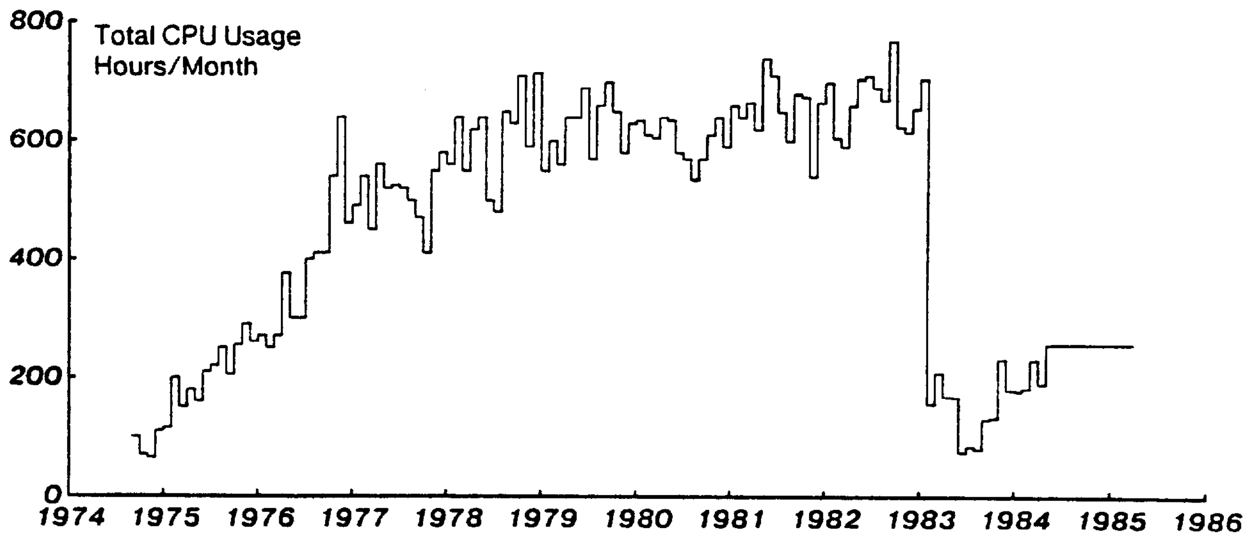

Figure 14: Total CPU Time Consumed by Month

Relative System Loading by Community

The SUMEX resource is divided, for administrative purposes, into three major communities: user projects based at the Stanford Medical School (*Stanford Projects*), user projects based outside of Stanford (*National AIM Projects*), and common system development efforts (*System Staff*). As defined in the resource management plan approved by the BRP at the start of the project, the available system CPU capacity and file space resources are divided between these communities as follows:

Stanford	40%
AIM	40%
Staff	20%

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

The monthly usage of CPU resources and terminal connect time for each of these three communities relative to their respective aliquots is shown in the plots in Figure 15 and Figure 16. As mentioned on page 293, these plots include both KI-10 and 2060 usage data.

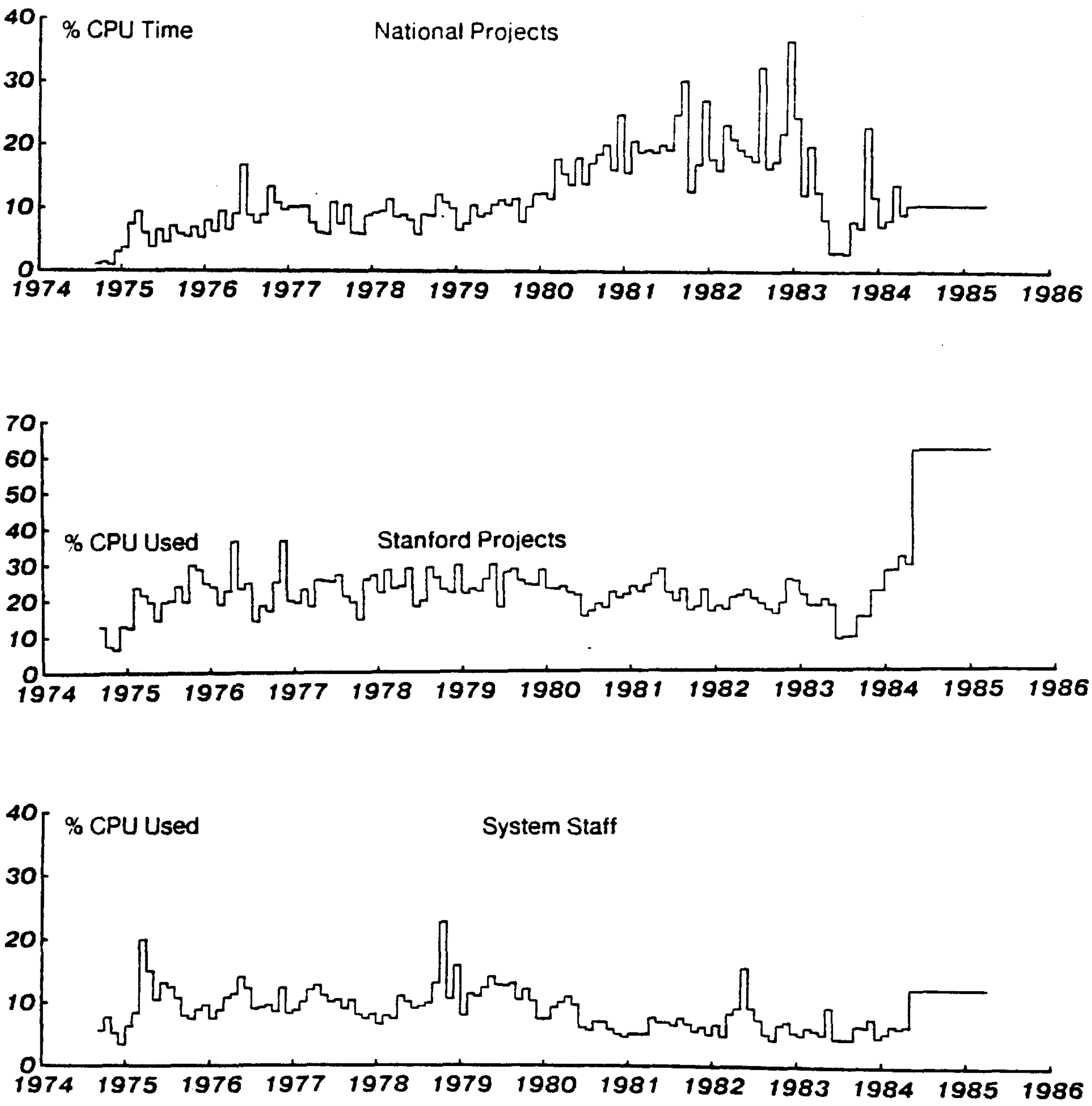

Figure 15: Monthly CPU Usage by Community

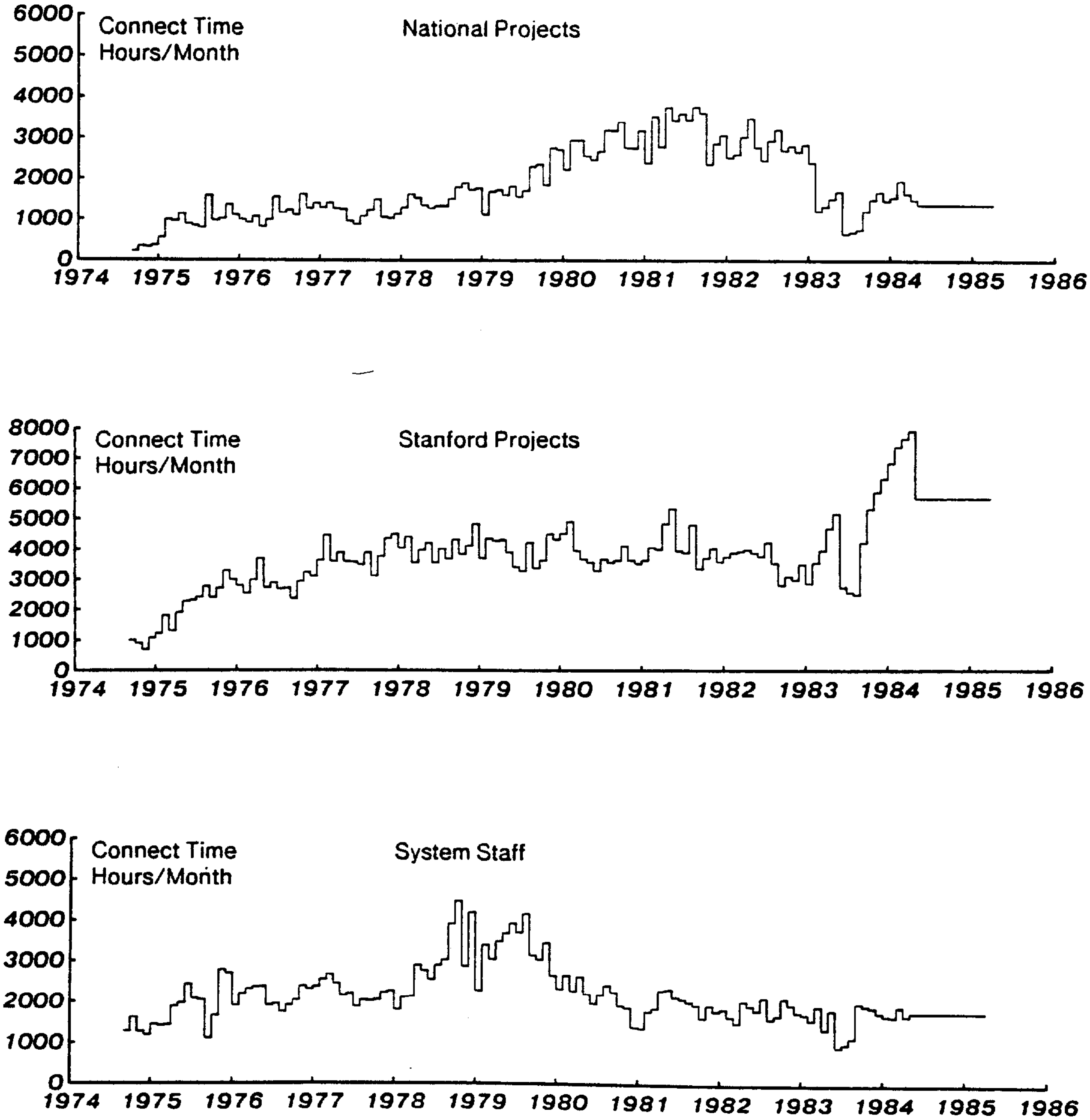

Figure 16: Monthly Terminal Connect Time by Community

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, 1984 and April 30, 1985, on the SUMEX-AIM DECsystem2060 system.

In the table following, entries include a text summary of the funding sources (outside of SUMEX-supplied computing resources) for currently active projects, total CPU consumption by project (Hours), total terminal connect time by project (Hours), and average file space in use by project (Pages, 1 page = 512 computer words). These data were accumulated for each project for the months between May, 1984 and May, 1985.

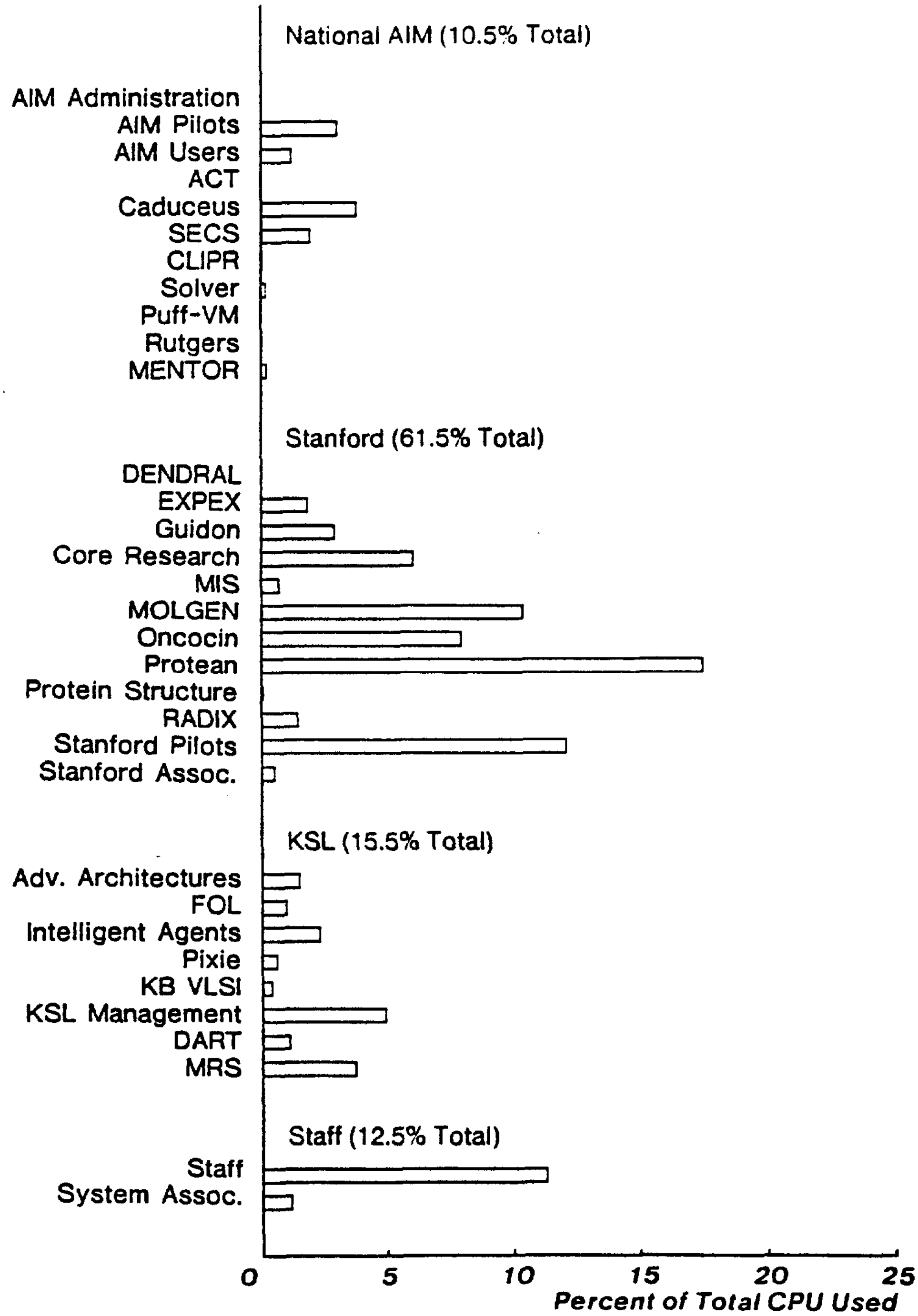

Figure 17: Cumulative CPU Usage Histogram by Project and Community

Resource Use by Individual Project - 5/84 through 4/85

National AIM Community	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) CADUCEUS "Clinical Decision Systems Research Resource" Jack D. Myers, M.D. Harry E. Pople, Jr., Ph.D. University of Pittsburgh NIH 5 R24 RR-01101-08 7/80-6/85 $1,607,717 7/84-6/85 $354,211 NIH 5 R01 LM03710-05 7/80-6/85 $817,884 7/84-6/85 $210,091 NIH New Invest 5 R23 LM03889-03 Gordon E. Banks, M.D. 4/82-3/85 $107,675 4/84-3/85 $35,975	86.72	1809.97	8028
2) CLIPR Project "Hierarchical Models of Human Cognition" Walter Kintsch, Ph.D. Peter G. Polson, Ph.D. University of Colorado NIMH 5 R01 MH-15872-14-16 (Kintsch) 7/84-6/87 $145,500 7/84-6/85 $40,500 NSF (Kintsch) 8/83-7/86 $200,000(*) IBM (Polson) David Kieras University of Arizona 1/85-12/85 $250,000(*)	1.14	119.94	129
3) SECS Project "Simulation & Evaluation of Chemical Synthesis" W. Todd Wipke, Ph.D. U. California, Santa Cruz NIHEHS ES02845-02 4/82-3/85 $257,801 4/84-9/85 $89,140 Evans & Sutherland Corp. Equipment gift Value $95,000 Stauffer Chemical Co. $6,000	45.14	5542.39	12230

4)	SOLVER Project "Problem Solving Expertise" Paul E. Johnson, Ph.D. William B. Thompson, Ph.D. University of Minnesota Control Data Corp. (Johnson) 1983-85 $90,000 Microelect. and Info. Ctr. Univ. of MN (Johnson, Thompson, Slagle, Wechsler, Yonas) 1984-1985 $500,000 NIH LM-00160 (Johnson, Connelly) 1984-1989 $712,573 McKnight Foundation (Johnson, Bailey) 1984-1985 $13,000 Dwan Family Fund, Univ. of MN Medical School (Johnson) 1985 $6,000	4.70	413.29	621
5)	MENTOR Project "Medical Evaluation of Therapeutic Orders" Stuart M. Speedie, Ph.D. University of Maryland Terrence F. Blaschke, M.D. Stanford University National Center for Health Services Research 1 R18 HS 05263 1/85-12/88 $485,134 1/85-12/85 $147,170	5.41	497.78	380
6)	*** [Rutgers-AIM] *** Rutgers Research Resource Artificial Intelligence in Medicine Casimir Kulikowski, Ph.D. Sholom Weiss, Ph.D. Rutgers U., New Brunswick NIH RR-02230-02 (Kulikowski, Weiss) 12/83-11/87 $3,198,075 12/84-11/85 $613,897	0.62	57.29	196
7)	AIM Pilot Projects	69.84	4292.54	3501
8)	AIM Administration	0.42	57.86	673
9)	AIM Users	27.88	3498.43	7135
		---------	----------	-------
	Community Totals	241.87	16289.49	32893

Stanford Community	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) GUIDON-NEOMYCIN Project Bruce G. Buchanan, Ph.D. William J. Clancey, Ph.D. Dept. Computer Science ONR/ARI N00014-79-C-0302 3/79-3/85 $683,892(*)	67.60	8225.93	6048
2) MOLGEN Project "Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing and Modification" Edward A. Feigenbaum, Ph.D. Peter Friedland, Ph.D. Charles Yanofsky, Ph.D. Depts. Computer Science/ Biology NSF MCS-8310236 (Feigenbaum, Yanofsky) 11/83-10/85 $270,836(*) 11/84-10/85 $131,621(*)	238.64	8358.21	11392
3) ONCOCIN Project "Knowledge Engineering for Med. Consultation" Edward H. Shortliffe, M.D., Ph.D. Dept. Medicine NIH RR-01613 7/83-6/86 $624,455 7/84-6/85 $222,511 NIH LM-04136 8/83-7/86 $211,851 8/84-7/85 $69,875 H.J. Kaiser Family Fdn. 7/83-6/86 $150,000 7/84-6/85 $50,000 NSF IST83-12148 Bruce G. Buchanan (Shortliffe) 3/84-2/87 $330,000(*) 3/84-2/85 $101,308(*) NIH 1 T32 LM07033 7/84-6/89 $903,718 7/84-6/85 $79,059 NIH 1 R23 LM04316 2/85-1/88 $107,441 2/85-1/86 $37,500	182.81	18869.06	16406

4)	PROTEAN PROJECT Oleg Jardetzky School of Medicine Bruce Buchanan Computer Science Department NSF PCM-84-02348 11/84-10/86 $100,000(*) 11/84-10/85 $50,000(*)	401.52	8539.01	13156
5)	RADIX Project "Deriving Medical Knowledge from Time Oriented Clinical Databases" Robert L. Blum, M.D. Gio C.M. Wiederhold, Ph.D. Depts. Computer Science/ Medicine NSF IST-8317858 (Blum) 3/84-3/86 $89,597(*) NIH LM-04334 (Wiederhold) 5/84-11/86 $291,192	33.23	2315.62	9168
6)	Stanford Pilot Projects	277.71	6545.02	5092
7)	Core AI Research	139.65	9447.97	10358
8)	Stanford Associates	11.40	1030.22	1127
9)	Medical Information Sciences	16.52	2561.42	974
		---------	----------	-------
	Community Totals	1369.08	65892.46	70901

KSL-AI Community	CPU (Hours)	Connect (Hours)	File Space (Pages)
For funding details please see page 105			
1) Advanced Architectures	34.45	11070.95	3313
2) FOL	22.61	781.19	1522
2) Intelligent Agent	53.25	6934.73	3205
3) Pixie	12.98	1989.63	1072
4) KB VLSI	8.47	1275.64	927
5) KSL Management	114.18	21341.80	15597
6) DART	25.05	1497.89	12677
7) MRS	86.40	9298.69	1950
	---------	----------	-------
Community totals	357.39	54190.52	40263

SUMEX Staff	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) Staff	261.44	21450.55	17051
2) System Associates	26.84	1809.75	4744
	---------	----------	-------
Community Totals	288.28	23260.30	21795

System Operations	CPU (Hours)	Connect (Hours)	File Space (Pages)
1) Operations	775.69	69589.10	131640
	=====	=====	=====
Resource Totals	3032.31	229221.87	297492

(*) Award includes indirect costs.

System Reliability

System reliability for the DECsystem 2060 has significantly improved in this past period. We have had very few periods of particular hardware or software problems. The data below covers the period of May 1, 1984 to April 30, 1985. The actual downtime was rounded to the nearest hour.

Table 1 : System Downtime Hours per Month - May 1984 through April 1985

13	1	16	5	9	17	1	N/A	26	9	8	9
May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr

Table 2 : System Downtime Hours per Month - May 1984 through April 1985

Reporting period	:	364 days, 19 hours, 13 minutes, and 25 seconds
Total Up Time	:	359 days, 11 hours, 32 minutes, and 18 seconds
PM Downtime	:	1 days, 6 hours, 8 minutes, and 1 seconds
Actual Downtime	:	4 days, 1 hours, 33 minutes, and 6 seconds
Total Downtime	:	5 days, 7 hours, 41 minutes, and 7 seconds
Mtbf	:	3 days, 14 hours, 16 minutes, and 31 seconds
Uptime Percentage	:	98.89

Network Usage Statistics

The plots in Figure 18 and Figure 19 show the monthly network terminal connect time for the TYMNET 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 (IP/TCP) in addition to the Arpanet, which converted to IP/TCP in January of 1983, it is no longer possible to distinguish between Arpanet usage and Internet usage on our 2060 system.

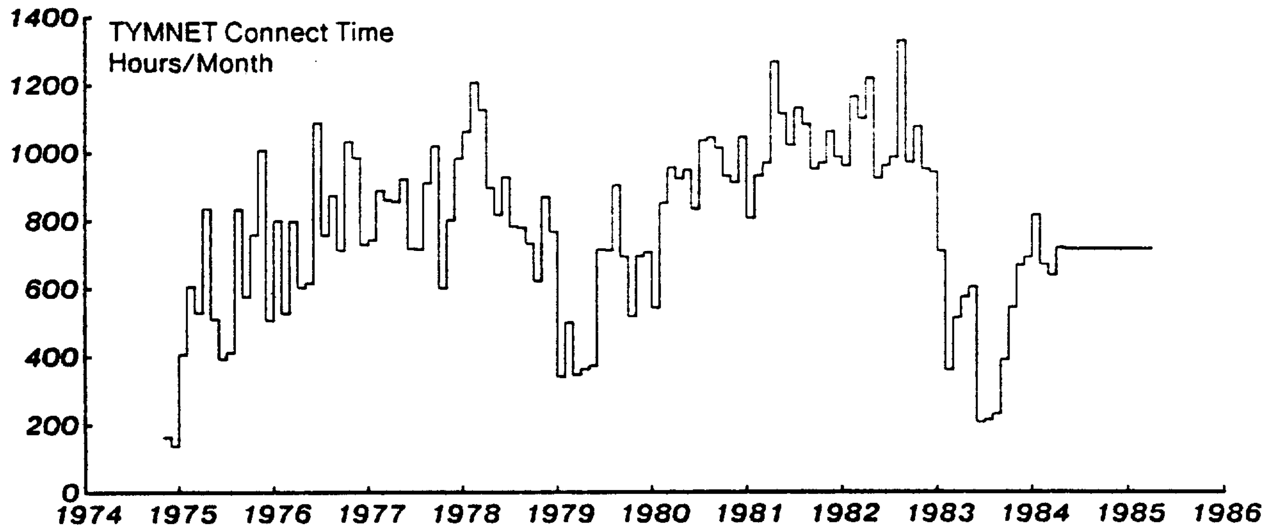

Figure 18: TYMNET Terminal Connect Time

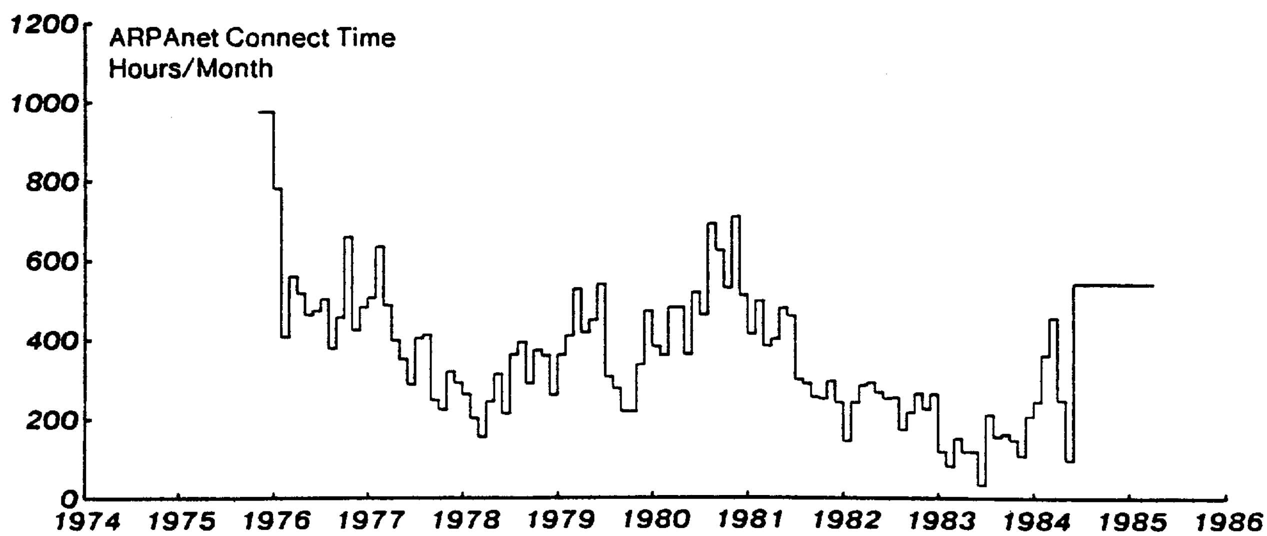

Figure 19: ARPANET Terminal Connect Time

Appendix C

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 Computer Science, TC135
Stanford University Medical Center
Stanford, California 94305
(415) 497-6970

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) 497-4879

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 Grp Chrmn)
Director, National Library of Medicine
8600 Rockville Pike
Bethesda, Maryland 02114
(617) 726-8311

MYERS, Jack D., M.D.
School of Medicine
Scaife Hall, 1291
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
(412) 624-2649

AIM Management Committee Membership

AIM Advisory Group:

MYERS, Jack D., M.D. (Chairman)
School of Medicine
Scaife Hall, 1291
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
(412) 624-2649

AMAREL, Saul, Ph.D.
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-3546

BAKER, William R., Jr., Ph.D. (Exec. Secretary)
10235 Gainsborough Road
Potomac, Maryland 20854

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) 497-4879

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.
Director, National Library of Medicine
8600 Rockville Pike
Bethesda, Maryland 02114
(617) 726-8311

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 20205
(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 Computer Science, TC135
Stanford University Medical Center
Stanford, California 94305
(415) 497-6970

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:

FEIGENBAUM, Edward A., Ph.D. (Chairman)
Heuristic Programming Project
Department of Computer Science
Margaret Jacks Hall
Stanford University
Stanford, California 94305
(415) 497-4879

LEVINTHAL, Elliott C., Ph.D.
Departments of Mechanical and Electrical Engineering
Building 530
Stanford University
Stanford, California 94305
(415) 497-9037

SHORTLIFFE, Edward H., M.D., Ph.D.
Principal Investigator - SUMEX
Medical Computer Science, TC135
Stanford University Medical Center
Stanford, California 94305
(415) 497-6970

Appendix D

Collaborative Project Abstracts

The following are brief abstracts of our collaborative research projects.

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) 497-1997 (CLANCEY@SUMEX-AIM)

Bruce G. Buchanan, Ph.D.
Computer Science Department
701 Welch Road
Stanford University
Palo Alto, California 94304
(415) 497-0935 (BUCHANAN@SUMEX-AIM)

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 consulation 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 will be used in the GUIDON2 family of instructional programs, being developed on D-machines.

REFERENCES

1. Clancey, W.J.: *Overview of GUIDON.* In A. Barr and E.A. Feigenbaum (Eds.), THE HANDBOOK OF ARTIFICIAL INTELLIGENCE, Vol. 2. William Kaufmann Assoc., Los Altos, CA, 1982. (Also to appear in J. of Computer-based Instruction)

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. In preparation. HPP-84-2.

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) 497-2413

Contact: Dr. Peter FRIEDLAND@SUMEX-AIM
(415) 497-3728

The goal of the MOLGEN Project is to apply the techniques of artificial intelligence to the domain of molecular biology with the aim of providing assistance to the experimental scientist. Previous work has focused on the task of experiment design. Two major approaches to this problem have been explored, one which instantiates abstracted experimental strategies with specific laboratory tools, and one which creates plans in toto, heavily influenced by the role played by interactions between plan steps. As part of the effort to build an experiment design system, a knowledge representation and acquisition package--the UNITS System, has been constructed. A large knowledge base, containing information about nucleic acid structures, laboratory techniques, and experiment-design strategies, has been developed using this tool. Smaller systems, such as programs which analyze primary sequence data for homologies and symmetries, have been built when needed.

New work has begun on scientific theory formation, modification, and testing. This work will be done within the domain of regulatory genetics. We plan to explore fundamental issues in machine learning and discovery, as well as construct systems that will assist the laboratory scientist in accomplishing his intellectual goals.

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

3. Stefik, M.J.: *An examination of a frame-structured representation system*, Proc. Sixth IJCAI, Tokyo, August, 1979, pp. 845-852.

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

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 - Room TC135
Stanford, California 94305
(415) 497-6979 (SHORTLIFFE@SUMEX-AIM)

Project Director: Dr. Lawrence M. Fagan

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 work is underway to transfer the program to professional workstations (rather than the central SUMEX computer) so that it can be implemented and evaluated at sites away from the University.

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 is designed for special high speed terminals and therefore cannot be tested or demonstrated via network connections. 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.

Stanford Project: PROTEAN Project

Principal Investigators: Oleg Jardetzky (JARDETZKY@SUMEX-AIM)
Nuclear Magnetic Resonance Lab, School of Medicine
Stanford University Medical Center
Stanford, California 94305

Bruce Buchanan, Ph.D. (BUCHANAN@SUMEX-AIM)
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

REFERENCES

1. Erman, L.D., Hayes-Roth, B., Lesser, V.R., Reddy, D.R.:*The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty*. ACM Computing Surveys 12(2):213-254, June, 1980.

2. Hayes-Roth, B.: *The Blackboard Architecture: A General Framework for Problem Solving?* Report HPP-83-30, Department of Computer Science, Stanford University, 1983.

3. 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.

4. Hayes-Roth, B.:*A Blackboard Architecture for Control*. Artificial Intelligence In Press, 1985.

5. Hayes-Roth, B. and Hewett, M.: *Learning Control Heuristics in BB1*. Report HPP-85-2, Department of Computer Science, 1985.

6. 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.

Stanford Project: RADIX -- DERIVING KNOWLEDGE FROM TIME-ORIENTED CLINICAL DATABASES

Principal Investigators:

Robert L. Blum, M.D.
Departments of Medicine
and Computer Science
Stanford University
Stanford, California 94305
(415) 497-9421 (BLUM@SUMEX-AIM)

Gio C.M. Wiederhold, Ph.D.
Department of Computer Science
Stanford University
Stanford, California 94305
(415) 497-0685 (WIEDERHOLD@SUMEX-AIM)

The objective of clinical database (DB) systems is to derive medical knowledge from the stored patient observations. However, 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.

The goal of the RADIX Project is to explore the usefulness of knowledge-based computational techniques in solving this problem of accurate knowledge inference from non-randomized, non-protocol patient records. Central to RADIX is a knowledge base (KB) of medicine and statistics, organized as a taxonomic tree consisting of frames with attached data and procedures. The KB is used to retrieve time-intervals of interest from the DB and to assist with the statistical analysis. Derived knowledge is incorporated automatically into the KB. The American Rheumatism Association DB containing records of 1700 patients is used.

SOFTWARE AVAILABLE ON SUMEX

RADIX--(excluding the knowledge base and clinical database) consists of approximately 400 INTERLISP functions. The following groups of functions may be of interest apart from the RADIX environment:

SPSS Interface Package -- Functions which create SPSS source decks and read SPSS listings from within INTERLISP.

Statistical Tests in INTERLISP -- Translations of the Piezer-Pratt approximations for the T,F, and Chi-square tests into LISP.

Time-Oriented Data Base and Graphics Package -- Autonomous package for maintaining a time-oriented database and displaying labelled time-intervals.

REFERENCES

Monograph

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

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.

Blum, R.L.: *Displaying clinical data from a time-oriented database.* Computers in Biology and Medicine 11(4):197-210, 1981.

Conference Proceeding

Blum, R.L.: *Modeling and encoding clinical causal relationships.* Proc. SCAMC83, IEEE, Baltimore, MD, October, 1983.

National AIM Project: CADUCEUS (formerly INTERNIST)

Principal Investigators:
Jack D. Myers, M.D. (MYERS@SUMEX-AIM)
Harry E. Pople, Ph.D. (POPLE@SUMEX-AIM)
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
Dr. Pople: (412) 624-3490
Dr. Myers: (412) 624-2649

The major goal of the CADUCEUS 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-off as a diagnostic and triage aid to physicians' assistants, rural health clinics, military medicine and space travel. In the design of CADUCEUS and its predecessor INTERNIST I, we have attempted to model the creative, problem-formulation aspect of the clinical reasoning process. The program employs a novel heuristic procedure that composes differential diagnoses, dynamically, on the basis of clinical evidence. During the course of a CADUCEUS or INTERNIST-1 consultation, it is not uncommon for a number of such conjectured problem foci to be proposed and investigated, with occasional major shifts taking place in the program's conceptualization of the task at hand.

SOFTWARE AVAILABLE ON SUMEX

Versions of INTERNIST are available for experimental use, but 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.

National AIM Project: CLIPR -- HIERARCHICAL MODELS OF HUMAN COGNITION

Principal Investigators: Walter Kintsch, Ph.D. (KINTSCH@SUMEX-AIM)
Peter G. Polson, Ph.D. (POLSON@SUMEX-AIM)
Computer Laboratory for Instruction in Psychological Research (CLIPR)
Department of Psychology
University of Colorado
Boulder, Colorado 80302
(303) 492-6991
Contact: Dr. Peter G. Polson (Polson@SUMEX-AIM)

The CLIPR Project is concerned with the modeling of complex psychological processes. It is comprised of two research groups. The prose comprehension group has completed a project that carries out the text analysis described by van Dijk & Kintsch (1983) yielding predictions of the recall and readability of that text by human subjects. The human-computer interaction group is developing a quantitative theory of that predicts learning, transfer, and performance for a wide range of computer-tasks, e.g. text editing.

SOFTWARE AVAILABLE ON SUMEX

A set of programs has been developed to perform the microstructure text analysis described in van Dijk & Kintsch (1983) and Kintsch & Greeno (1985). The program accepts a propositionalized text as input, and produces indices that can be used to estimate the text's recall and readability.

REFERENCES

1. Fletcher, R. C. Understanding and solving word arithmetic problems: A computer simulation. Technical Report NO. 135, Institute of Cognitive Science, Colorado, 1984.

2. Kieras, D.E. and Polson, P.G.: *The formal analysis of user complexity.* Int. J. Man-Machine Studies, In Press.

3. Kintsch, W. and van Dijk, T.A.: *Toward a model of text comprehension and production.* Psychological Rev. 85:363-394, 1978.

4. Kintsch, W. and Greeno, J.G.:*Understanding and solving word arithmetic problems*. Psychological Review, 1985, 92, 109-129.

5. Polson, P.G. and Kieras, D.E.: *A formal description of users' knowledge of how to operate a device and user complexity.* Behavior Research Methods, Instrumentation, & Computers, 1984, 16, 249-255.

6. Polson, P.G. and Kieras, D.E.: *A quantitative model of the learning and performance of text editing knowledge.* Proceedings of the CHI 1985 Conference on Human Factors in Computing. San Francisco, April 1985.

7. van Dijk, T.A. and Kintsch, W.:*STRATEGIES OF DISCOURSE COMPREHENSION*. Academic Press, New York, 1983.

8. Young, S. A theory and simulation of macrostructure. Technical Report No. 134, Institute of Cognitive Science, Colorado, 1984.

9. Walker, H.W. & Kintsch, W. Automatic and strategic aspects of knowledge retrieval. Cognitive Science, 1985, 9, 261-283.

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 will use principles of artificial intelligence to create 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 will be automatically acquired through interfaces to a hospital information system. This data will be 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 will be fed back to the responsible physicians to assist future decision making.

Specific objectives of this proposal include:

1. Implement a prototype computer-based expert system to continuously monitor in-patient drug therapy. It will use 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. Design and begin implementation of an evaluation of 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 the proposed project will build 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.

National AIM Project: SOLVER -- PROBLEM SOLVING EXPERTISE

Principal Investigators:

Paul E. Johnson, Ph.D., School of Management and
Center for Research in Human Learning
205 Elliott Hall
University of Minnesota
Minneapolis, Minnesota 55455
(612) 376-2530 (PJOHNSON@SUMEX-AIM)

William B. Thompson, Ph.D.
Department of Computer Science
136 Lind Hall
University of Minnesota
Minneapolis, Minnesota 55455
(612) 373-0132 (THOMPSON@SUMEX-AIM)

The Minnesota SOLVER project focuses upon the development of strategies for discovering and representing the knowledge and skill of expert problem solvers. Although in the last 15 years considerable progress has been made in synthesizing the expertise required for solving complex problems, most expert systems embody only a limited amount of expertise. What is still lacking is a theoretical framework capable of reducing dependence upon the expert's intuition or on the near exhaustive testing of possible organizations. Our methodology consists of: (1) extensive use of verbal thinking aloud protocols as a source of information from which to make inferences about underlying knowledge structures and processes; (2) development of computer models as a means of testing the adequacy of inferences derived from protocol studies; (3) testing and refinement of the cognitive models based upon the study of human and model performance in experimental settings. Currently, we are investigating problem-solving expertise in domains of medicine, financial auditing, management, and law.

SOFTWARE AVAILABLE ON SUMEX

A redesigned version of the Diagnoser simulation model, named Galen, has been implemented on SUMEX.

REFERENCES

1. Johnson, P.E., Moen, J.B., and Thompson, W.B.: *Garden Path Errors in Medical Diagnosis.* IN Bloc, L. and Coombs, M.J. (Eds.), COMPUTER EXPERT SYSTEMS, Springer-Verlag (in press).

2. Johnson, P.E., Johnson, M.G., and Little, R.K.: *Expertise in trial advocacy: Some considerations for inquiry into its nature and development*, Campbell Law Review, (in press).

3. Johnson, P.E., "The Expert Mind: A New Challenge for the Information scientist," in *Beyond Productivity: Information System Development for Organizational Effectiveness*, Th. M. A. Bemelmans (editor), Elsevier Science Publishers B. V. (North-Holland), 1984.

4. Johnson, P.E.: *What kind of expert should a system be?* J. Medicine and Philosophy, 8:77-97, 1983.

5. Johnson, P.E., Duran, A., Hassebrock, F., Moller, J., Prietula, M., Feltovich, P. and Swanson, D.: *Expertise and error in diagnostic reasoning*. Cognitive Science 5:235-283, 1981.

6. Thompson, W.B., Johnson, P.E. and Moen, J.B.: *Recognition-based diagnostic reasoning*. Proc. Eighth IJCAI, Karlsruhe, West Germany, August, 1983.

Stanford Pilot Project: THE COMPUTER-AIDED MEDICAL DECISION ANALYSIS (CAMDA) PROJECT

Co-Principal Investigator: Samuel Holtzman
Ronald A. Howard
Department of Engineering-Economic Systems
Stanford University
Stanford, California 94305

Contact: Samuel Holtzman(HOLTZMAN@SUMEX-AIM)
(415) 497-0486

The CAMDA project is a program of research in the area of medical decision making. The main focus of this effort is to combine decision analysis and artificial intelligence to develop systems that support medical decisions.

Nearly two decades of experience in the application of decision analysis to problems in industry and government have shown that the technique constitutes an extremely helpful tool for making difficult choices. The potential benefit of decision analysis is particularly great when choices must be made in the presence of uncertainty and when the stakes involved are high. This situation is common in medical decisions.

Partly as a result of the high cost of an individual decision analysis, and partly due to the inherent complexity of making choices which involve outcomes such as pain and death, medical decision analysis has remained essentially within the realm of the academic community. Therefore, the majority of patients and physicians have been deprived of the benefits of this powerful technique.

Expert system technology makes it possible to bring decision analysis to the medical community in general. By providing a sophisticated modeling methodology, expert systems allow the process of decision analysis (within a specific medical context) to be formalized with sufficient accuracy to make much of the analysis amenable to computer automation. The resulting CAMDA systems could provide an attractive alternative to unaided decision making, and to the usually unaffordable option of analyzing medical decisions individually. Furthermore, these systems can help decision makers think more clearly about the difficult issues they face by providing them with a means to experiment with the logical consequences of their assumptions and preferences.

A major focus of our research effort is the development of RACHEL, an intelligent decision system for infertile couples. The field of infertility was chosen for several reasons, including the prevalence of the condition, the complexity of the values that are usually attached to the possible outcomes in this field, the rapidly growing set of available tests and treatments, and the time-dependent nature of the human reproductive process.

As part of the development of RACHEL, a substantial portion of the current CAMDA effort is aimed at the development of a general computer-based aid for medical decision analysis, which could be used in other medical decision domains.

REFERENCES:

1. Holtzman, S.: *A Decision Aid for Patients with End-Stage Renal Disease*, Department of Engineering-Economic Systems, Stanford University, Stanford, California, 1983.

2. Holtzman, S.:*On the Use of Formal Methods for Decision Making*, Department of Engineering-Economic Systems, Stanford University, 1985.

3. (*) Holtzman, S.:*Intelligent Decision Systems*, Ph.D. Dissertation, Department of Engineering-Economic Systems, Stanford University, 1985.

Stanford Project: REFEREE Project

Principal Investigators:

Bruce G. Buchanan, Ph.D.
Computer Science Department
701 Welch Road
Stanford University
Palo Alto, California 94304
(415) 497-0935
(BUCHANAN@SUMEX-AIM)

Byron W. Brown Ph.D.
Department of Biostatistics
Stanford University Medical Center
Stanford, California 94305
(BWROWN@SUMEX-AIM)

Daniel E. Feldman, M.D., Ph.D.
Department of Medicine
Stanford University Medical Center
Stanford, California 94305
(DFELDMAN@SUMEX-AIM)

The goal of this project is two-fold: (a) use existing AI methods to implement an expert system that can critique medical journal articles on clinical trials, and (b) in the long term, develop new AI methods that extract new medical knowledge from the clinical trials literature. In order to accomplish (a) we are building the system in three stages.

1. System I will assist in the evaluation of the quality of a single clinical trial. The user will be imagined to be the editor of a journal reviewing a manuscript for publication, but the program will be tested on a variety of readers, including clinicians, medical scientists, medical and graduate students, and clerical help.

2. System II will assist in the evaluation of the effectiveness of the treatment or intervention examined in a single published clinical trial. The user will be imagined to be a clinician interested in judging the efficacy of the treatment being tested in the trial.

3. System III will assist in the evaluation of the effectiveness of a single treatment examined in a number of published clinical trials.

National AIM Project: Computer-Aided Diagnosis of Malignant Lymph Node Diseases (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)

Lawrence M. Fagan, M.D., Ph.D.
Department of Medicine
Stanford University Medical Center - Room TC135
Stanford, California 94305
(415) 497-6979 (FAGAN@SUMEX-AIM)

We are building a computer program, called PATHFINDER, to assist in the diagnosis of lymph node pathology. The project is based at the University of Southern California in collaboration with the Stanford University Medical Computer Science Group. A pilot version of the program provides diagnostic advice on 80 common benign and malignant diseases of the lymph node based on 150 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.

SOFTWARE AVAILABLE ON SUMEX

PATHFINDER-- A version of the PATHFINDER program is available for experimentation on the DEC 2060 computer. This version is a pilot version of the program, and therefore has not been completely tested.

National AIM Project: RXDX Project

Principal Investigators: Robert Lindsay, Ph.D.
Michael Feinberg, M.D., Ph.D.
Manfred Kochen, Ph.D.
University of Michigan
Ann Arbor, Michigan

We are developing a prototype expert system that could act as a consultant in the diagnosis and management of depression. Health professionals will interact with the program as they might with a human consultant, describing the patient, receiving advice, and asking the consultant about the rationale for each recommendation. The program uses a knowledge base constructed by encoding the clinical expertise of a skilled psychiatrist in a set of rules and other knowledge structures. It will use this knowledge base to decide on the most likely diagnosis (endogenous or nonendogenous depression), assess the need for hospitalization, and recommend specific somatic treatments when this is indicated (e.g., tricyclic antidepressants). The treatment recommendation will take into account the patient's diagnosis, age, concurrent illnesses, and concurrent treatments (drug interactions).

References

1. Barr, Avron, Paul R. Cohen and Edward A. Feigenbaum. *The Handbook of Artificial Intelligence, Volumes I, II, and III.* William Kaufmann, Inc., Los Altos, CA, 1981 and 1982.

2. Bennett, James S. and Robert S. Engelmore. "SACON: A Knowledge-Based Consultant for Structural Analysis." *IJCAI Proceedings 6* (1979), 47 - 49.

3. Bobrow, D., ed.. *Artificial Intelligence, An International Journal: Special Volume on Qualitative Reasoning about Physical Systems.* North Holland Publishing Company, Amsterdam, 1984.

4. Bryson, Y.J., Dillon, M., Lovett, M., et al. "Treatment of first episodes of genital herpes simplex infection with oral acyclovir." *New England Journal of Medicine 308* (1983), 916-921.

5. Buchanan, B.G. Mechanizing the Search for Explanatory Hypotheses. Tech. Rept. KSL-82-28, Knowledge Systems Laboratory, Stanford University, 1982.

6. Buchanan, B.G., and Shortliffe, E.H.. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley Publishing Company, Reading, MA, 1984.

7. Buchanan, B.G., G.L. Sutherland, and E.A. Feigenbaum. Rediscovering some problems of artificial intelligence in the context of organic chemistry. In *Machine Intelligence 5*, B. Meltzer and D. Michie, Eds., Edinburgh University Press, Edinburgh, Scotland, 1969, pp. 253-280.

8. Buchanan, B. G., Mitchell, T. M., Smith, R. G. and Johnson, C. R. Jr. "Models of Learning Systems." *Encyclopedia of Computer Science and Technology 11* (1978), 24-51. also Stanford report STAN-CS-79-692

9. Cheriton, David R. "The V Kernel: A Software Base for Distributed Systems." *IEEE 1*, 2 (April 1984).

10. Clancey, W.J. Heuristic Classification. Tech. Rept. KSL-85-5, Knowledge Systems Laboratory, Stanford University, March, 1985.

11. Coulter, C. L. "Research Instrument Sharing." *Science 201*, 4354 (1978).

12. Culliton, Barbara J. "Information as a 'Cure' for Cancer." *Science 227* (1985).

13. Davis, R. "Interactive Transfer of Expertise: Acquisition of New Inference Rules." *Artificial Intelligence 12* (1979), 121-158.

14. Davis, R. "Diagnostic Reasoning Based on Structure and Behavior." *Artificial Intelligence 24* (December 1984).

15. Emerson, J.D., Graham, A.C. "Use of statistical analysis in the New England Journal of Medicine." *New England Journal of Medicine 309* (1983), 1332-1337.

16. Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, D.R. "The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty." *Computing Surveys 12* (1980), 213-253.

17. Fagan, L.M., Kunz, J.C., Feigenbaum, E.A., and Osborn, J.J. Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit. Proceedings of the 6th International Joint Conferences on Artificial Intelligence, Tokyo, 1979, pp. 260-262.

18. Feinstein, A.R., Horwitz, R.I., Spitzer, W.O., Battista, R.N. "Coffee and Pancreatic Cancer." *JAMA 246* (1981), 957-961.

19. Fu, Li-Min. *Learning Object-level and Meta-level Knowledge for Expert Systems.* Ph.D. Th., Stanford University, 1985.

20. Genesereth, M.R. A Meta-Level Representation System. Tech. Rept. HPP83-28, Stanford University, Knowledge Systems Laboratory, 1983. The Report Consists of Four: HPP80-24, HPP82-27, HPP83-26, and HPP84-1

21. Genesereth, M.R. "The Role of Design Descriptions in Automated Diagnosis." *Artificial Intelligence 24* (December 1984).

22. Goldin, S. E., and Hayes-Roth, B. Individual differences in planning processes. Tech. Rept. N-1488-ONR, Santa Monica, Ca.: Rand Corporation, 1980.

23. Haggerty, J. REFEREE and RULECRITIC: Two prototypes for assessing the quality of a medical paper. Master Th., Stanford University, 1984.

24. Hart, P.O., Duda, R.O., Einaudi, M.T. "PROSPECTOR -- A Computer-based Consultation System for Mineral Exploration." *Mathematical Geology 10*, 5 (1978).

25. Hayes-Roth, B. Flexibility in executive processes. Tech. Rept. N-1170-ONR, Santa Monica, Ca.: Rand Corporation, 1980.

26. Hayes-Roth, B. The blackboard architecture: A general framework for problem-solving? Tech. Rept. HPP-83-30, Stanford, Ca.: Stanford University, 1983.

27. Hayes-Roth, B. BB1: An architecture for blackboard systems that control, explain, and learn about their own behavior. Tech. Rept. HPP-84-16, Stanford, Ca.: Stanford University, 1984.

28. Hayes-Roth, B. "A blackboard architecture for control." *Artificial Intelligence Journal in press* (1985).

29. Hayes-Roth, B., and Hewett, M. Learning Control Heuristics in a Blackboard Environment. Tech. Rept. HPP-85-2, Stanford, Ca.: Stanford University, 1985.

30. Hayes-Roth, B., Hayes-Roth, F., Rosenschein, S., and Cammarata, S. "Modelling planning as an incremental, opportunistic process." *Proceedings of the International Joint Conference on Artificial Intelligence 6* (1979), 375-383.

31. Hayes-Roth, F., Waterman, D. A. & Lenat, D. B. (Eds.). *Building Expert Systems.* Addison-Wesley, Reading, MA, 1983.

32. Hewitt, C., P. Bishop, and R. Steiger. A Universal Modular ACTOR Formalism for Artificial Intelligence. IJCA13, 1973, pp. 235-45.

33. 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. KSL 85-21. Submitted for publication, May 1985

34. 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 17-14, 1984, 1984.

35. 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. KSL 85-22. Submitted for publication, March 1985

36. Kunz, J.C., Fallat, R.J., McClung, D.H., Osborn, J.J., Votteri, B.A., Nii, H.P., Aikins, J.S., Fagan, L.M., Feigenbaum, E.A. A physiological rule-based system for interpreting pulmonary function test results. Proceedings of Computers in Critical Care and Pulmonary Medicine, 1979, pp. 375-379.

37. Keith A. Lantz et. al. *V - System 4.1 Reference Manual.* December 1, 1983. Computer Systems Laboratory, Stanford University.

38. K. A. Lantz, K. D. Gradischnig, J. Feldman, and R. Raschid. "Rochester's Intelligent Gateway." *Computer 10*, 15 (October 1985), 54-68.

39. Keith A. Lantz, David R. Cheriton, and William I. Nowicki. Third Generation Graphics for Distributed Systems. Computer Systems Laboratory. Stanford University, December 21, 1982

40. P.J. Leach et al. "The Architecture of an Integrated Local Network." *IEEE Journal on Software and Applications ?*, SAC-1,5 (November 1983), 842-857.

41. Lederberg, J. "Digital Communications and the Conduct of Science: The New Literacy." *Proceedings of the IEEE 66*, 11 (1978).

42. Levy, R.I. "Current status of cholesterol controversy." *Am J Med 74* (1983), 1-4.

43. Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J.. *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project*. McGraw-Hill, New York, 1980.

44. MacMahon, B., Yen, S., Trichopoulos, D., et al. "Coffee and cancer of the pancreas." *New England Journal of Medicine 304* (1981), 630-633.

45. Masinter, Larry M. Interlisp VAX: A Report. Tech. Rept. HPP-81-14 or STAN-CS-81-879, Heuristic Programming Project, Stanford University, 1981.

46. McCarthy, J.M. "Circumscription -- A Form of Nonmonotonic Reasoning." *Artificial Intelligence 13* (April 1980).

47. McDermott, J. "R1: The Formative Years." *The AI Magazine 2*, 2 (1981). summer

48. McDermott, J. "R1: A rule-based configurer of computer systems." *Artificial Intelligence 9*, 1 (1982), 39-88.

49. Miller, P.L. "ATTENDING: Critiquing a Physician's Management Plan." *IEEE Transactions on Pattern Analysis and Machine Intelligence 5* (1983), 449-461.

50. Musen, M., Langlotz, C., Fagan, L., Shortliffe, E.H. Motivations for Restructuring a Rule-based Consultation System. Proceedings of the American Association for Medical Science and Informatics Congress, 1985. KSL 85-17

51. Newell, A. and Sproull, R. F. "Computer Networks - Prospects for Scientists." *Science 215*, 4534 (1978).

52. Newell, A. Reasoning, problem solving and decision processes: The problem space as a fundamental category. In *Attention and Performance VIII*, R. Nickerson, Ed.,Erlbaum, Hillsdale, NJ, 1980.

53. Newell, A. & Simon, H. A.. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, 1972.

54. Nii, H.P. and Aiello, N. AGE: A Knowledge-based Program for Building Knowledge-based Programs. Proceedings of IJCAI 6, 1979, pp. 645-655.

55. Nii, H. P., E. A. Feigenbaum, J. J. Anton, and A. J. Rockmore. "Signal-to-Symbol Transformation: HASP/SIAP Case Study." *AI Magazine 3*, 2 (spring 1982), 23 - 35.

56. Nilsson, N.. *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA, 1980.

57. Novak, Gorden. *GLISP User's Manual*. Heuristic Programming Project, Stanford University, Stanford Ca. 94305, 1982.

58. Nowicki, William I. *Partitioning of Function in a Distributed Graphics System.* Ph.D. Th., Stanford University, 1985.

59. Price, D.S. The development and structure of the biomedical literature. In *Coping with the Biomedical Literature*, Praeger, New York, 1981.

60. R. F. Rashid and G. G. Robertson. Accent: A Communication Oriented Operating System Kernel. SIGOPS Operating Systems Review 15(5), Eigth Symposium on Operating Systems Principals, December, 1981, pp. 64-75.

61. Rich, E.. *Artificial Intelligence*. McGraw-Hill, New York, 1983.

62. Rout, L. "Fast medical advances give doctors troubles in staying up to date." *Wall Street Journal 104* (1981), 1, 15.

63. Sackett, D.L. "How to read clinical journals: I. Why to read them and how to start reading them critically." *Can Med Assoc J 124* (1981), 555-558.

64. Shafer, G.. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, New Jersey, 1976.

65. Shortliffe, Edward H.. *Computer-Based Medical Consultation: MYCIN.* American Elsevier, New York, 1976.

66. Smith, Reid G., and Friedland, Peter. Unit Package User's Guide. Tech. Rept. KSL-80-28, Knowledge Systems Laboratory, Stanford University, December, 1980.

67. Smith, Reid G.. *A Framework for Distributed Problem Solving.* UMI Research Press, Ann Harbor, Michigan, 1981.

68. Van Melle, W., Scott, A.C., Bennett, J.S., and Peairs, M.A.S. The EMYCIN Manual. Tech. Rept. KSL-80-11, Knowledge Systems Laboratory, Stanford University, 1980. Superseded by KSL-81-16

69. Winston, P.. *Artificial Intelligence, 2nd ed.* Addison-Wesley, Reading, MA, 1984.

70. Zwaenepoel, Willy. *Message Passing on a Local Network.* Ph.D. Th., Stanford University, 1984.