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 resour:es 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, WJ:*Heuristic classification.* Knowledge Systems Laboratory Report KSL 85-5, Computer Science Dept., Stanford University, March 1985.
3. Richer, M., and Clancey, WJ.: *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, WJ.: *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.