structures from their images, the difficulty of human discrimination of textures and boundaries, and the difficulty of remembering the diagnostic search tree, the performance of medical specialists in diagnosing these images may be significantly aided by computers. Thus, for our project, computers provide the following basic functions:

1. Detection
2. Visualization
3. Consultation

We shall describe our computer algorithms for detecting tumors, finding the boundaries of the radiographic images of lungs, and the reconstruction of anatomic structures from their projection, and optimal search through decision tree for the analysis and diagnosis of medical images. We shall discuss our present approach to knowledge-based consultation for the diagnosis of chest radiographs.

3.11 AIMDS and Its Use in BELIEVER

N.S. Sridharan
Rutgers University

One of the central concerns in AI today is to develop a framework for the exploration of interpretation processes that are guided by one's expectations where a set of assimilatory schemata guide the integration of input with expectations. The input can be scenes for vision, signals for speech and action sequences for plan recognition. The interpretation process produces a network of instantiated forms of these schemata forming a bridge between the input and expectations. The formation of the bridge typically involves intermediate levels of representation. At the previous workshop we reported on AIMDS focusing on the language for specifying the structure of such schemata along with rules of consistent instantiation. We have implemented now the "Hypothesize and Revise" paradigm by developing procedures for setting up expectation structures, for matching observation against expectation and for resolution of conflict found by the match. The essence of this paradigm lies in interpreting the input by maintaining a few, often only one, hypothesis and expectation about the input. Further elaboration on this interpretation is directed by the conflicts found between observation and expectation, which trigger constructive processes of hypothesis revision and repair. This paradigm has potential payoff in that an explicit search of the space of alternative interpretations is avoided. This issue is being explored in the problem of plan recognition in BELIEVER. We have developed Action, Plan, Expectation Structure and Interpretation as
formally stated concepts and have defined logical rules for their composition. We are presently developing hypothesis revision rules and are exploring the specification of strategy for goal and plan recognition. These new developments will be discussed in the presentation by means of an example.

3.12 SECS Chemical Synthesis Project

Principal Investigator: W. Todd Wipke

Coworkers: S. Krishnan
Martin Huber
Dan Dolata
Glenn Ouchi

This talk will describe recent progress and future directions for the chemical synthesis program SECS. The purpose of the program is to help synthetic chemists design syntheses of complex biomolecules. Our recent work has been directed toward strategic control of the program by the chemist so essentially the program will only develop solutions which are likely to be of interest to the chemist. We have also studied strategies based on symmetry, and algorithms for symmetry recognition. The database of chemical reactions continues to grow and be improved both by the Santa Cruz group and by various collaborators. We are searching for methods to facilitate the control of our knowledge base.

Many improvements have been made to the SECS program to speed interaction over remote lines, to simplify use of the program, and to facilitate feedback from users to builders.

A new project was initiated for computer-aided prediction of metabolites for carcinogenicity studies. The fundamental goal is, given a specific molecule and a library of metabolic transformations, generate all possible metabolites of the compound. We have also explored the field of biosynthesis of marine natural products again by using a special library of biological reactions.

Summaries of Working Group Discussions
4.1 Guidelines for Designing Knowledge Representation

Saul Amarel - Moderator

Dr. Amarel summarized. The goal of the discussion was to set up guidelines for choosing a knowledge representation for a specific task. Researchers have accumulated considerable experience to do this in applying AI to a variety of task domains. These domains include: speech understanding, clinical medicine, chemistry, signal processing, and education.

Sridharan pointed out that the "fit" between a representation and a particular task requires that the system designer analyze the properties of the problem and use these properties to guide the choice of representation. A factor to consider when choosing a representation is the nature of the task as defined by performance. That is, is the task one of problem-solving, question-answering or search?

Other factors to consider are the acquisition, management and explanation of the knowledge. Should the explanation benefit the system designer for example in debugging the system, or should the explanation benefit the user? If so, should it be explanation for the purpose of instruction or for the purpose of justifying the design or the solution? One should be able to select which details are of use in the explanation and make a distinction between inside knowledge which benefits the system designer and outside knowledge which benefits the user. This raises the question of what traces of reasoning the user should be given. Should the explanation include only those traces which lead to the correct solution, or should it also include the rejected paths?

Sridharan described two open problems:

1. How can system designers set up a representation to insure the freedom to experiment with the control strategies and process definitions, and not be confined to a specific control strategy?

2. When a new rule is entered into the system, how does the designer insure that the system does not have that rule already?

Various representations were discussed: straight code, semantic nets and frames. Discussion focused on the question of using straight code rather than using code in addition to a packaged, rule-like structured system which would provide clarity from the outset and would facilitate building on others' work.

Key issues regarding semantic nets were: the necessity for
making a distinction between inferencing with a semantic net and creating a semantic net; also, updating semantic nets with regard to the problem of acquiring information in one form and compiling it into another.

Ed Feigenbaum suggested a strategy for knowledge engineering. In order to insure credibility and trust among users of the system, it is important to facilitate the transfer of their knowledge into the system. The system design should be transparent and oriented toward the user. The representations should be simple and the line of reasoning should be comprehensive and of benefit the user. The results should be presented in a language and format that is comfortable for the user and appropriate to the domain. Designers should aim for a system that is self-explanatory. He cited DENDRAL as one such system. In its design, the particular representation was preferred because it is natural and in the language of the chemist. The advantage to this was that eventually, chemists were able to take over the maintenance and development of the system.
4.2 Distributed Data Base Systems and Networks

Bruce McCormick - Moderator

Bruce McCormick listed the following issues:

1) Who has the traditional data base systems: The National Library of Medicine has a catalog of such systems. However, transferring the information is made difficult by the incompatibility of various formats.

2) Distributed data bases: For reasons of confidentiality, security, and reliability, the use of distributed data bases seems appropriate. Some data bases however, like the March of Dimes data base for research on birth defects, ought to be centralized.

3) Access to data bases: The traditional view is that data base access is done by asking a standard form of questions, and receiving a simple response in return. The possibility of browsing interactively through a data base should be explored as a part of the process of formulating questions.

4) Pictorial presentation of a data base: Often it is desirable to have a way of showing to a user what the structure of a given data base is. Is there a way to make a picture of a data base so that common parts are amalgamated into a simplified presentation? Such presentations could use color and interactive graphics to give a sense of the structure of a complex data base.

5) What, if any, are the differences between data bases and knowledge bases: Just as data bases ought to be movable from one application to another, so also should knowledge bases be movable from one interpretive system to another, such as the INTERNIST knowledge base of medical facts. Perhaps knowledge bases are not different conceptually from traditional data bases.

Members of the audience took issue with this last point. Ed Feigenbaum stated that it was "cutraneous" to suggest that knowledge bases and data bases are the same. Raj Reddy pointed out that they
differ in the way the information is queried and interpreted. Some participants in the discussion felt that AI experts should be involved more directly with medical practice to fully understand the problems of medical record keeping and accessing of medical information. The need for "hybrid" hardware systems was emphasized as a way of realizing such multiple purposes as clinical practice and research. A PDP11-class machine used for clinical practice for example, might be connected to a PDP10-class machine used for research.

4.3 Hardware/Software Gap in Knowledge-based System Design

Saul Levy reported on the discussion.

Most of the research and development of ATM systems is done in LISP on a PDP10. The way in which this work should be exported is the source of controversy. Should the systems be exported in LISP or recoded in another language? Proponents of LISP argue that the only problem with LISP is that it is too slow, but new hardware, for example LISP machines, and new faster technologies will enable the programs to run fast enough for production use. LISP systems can be speeded up in various ways. For example, it is not necessary to have a complete LISP system. A minimal cut-down system is sufficient. The opposing view is that in addition to being slow, LISP is not a good language for transferring programs. Recoding should be done in a language that facilitates transfer, such as MAINSAIL. Todd Wipke suggested FORTRAN as the language for recoding since all large programs that have been transported successfully were in FORTRAN. John S. Brown pointed out however, that it is easier to move a program from one LISP to another than from one manufacturer's FORTRAN to another.

4.4 Methods of Plausible Reasoning

Peter Szolovits - Moderator

Dr. Szolovits listed the following concerns of the discussion:

1) Collaborators: Medical collaborators should not serve solely as knowledge sources to be "squeezed dry". A good medical collaborator is one who is as interested in the processes which underlie his own reasoning as the system designer, thereby making the benefits of the collaboration mutual.
2) Knowledge Acquisition: It would be ideal if a system could acquire knowledge by reading medical textbooks and talking to doctors. Considerable research is needed in the area of representation before this could happen and before problems of acquisition in general, can be solved.

3) Hypothesis Formation and Reasoning: It is not clear what a hypothesis is, or how doctors reason. Experience in production systems reveals the difficulties of acquiring rules when the experts express their knowledge in terms of flowcharts. This may reflect the computing methods of the previous generation. It may be reasonable to use straight code if this suits the expert's method of reasoning.

4) Probabilistic Reasoning: Probabilistic reasoning is "dangerous" since certain assumptions have to be made. Under one set of assumptions a piece of evidence will have a certain strength of support for a hypothesis, while under different but equally reasonable assumptions it will have a different strength of support for the same hypothesis.

4.5 Problems of Scientific Communication/Collaboration Over Networks

Murray Turoff - Moderator

The SUMEX-ATM community has a great deal of experience in communicating over a network, however making scientific work available electronically within the community is not proceeding well, the problem being people are not making it happen. Tools like electronic bulletin boards at SUMEX have been developed for the purpose of facilitating scientific collaboration, but they are not being used. Rather, the computer network is being used as a message carrier. This raises the question: Is face to face interaction necessary for scientific collaboration?

There are two possible solutions to this problem: (1) to have a "facilitator" or a "pest" whose responsibility is to prompt communication among researchers using the network, and (2) to set up tasks which need scientific communication in order to get done, for example, the publication of the Workshop Proceedings.
Sridharan commented that the computer helps in routine communication so that time spent in face to face interaction can be devoted almost entirely to scientific research.
Each panelist was asked to discuss aspects of the design and implementation of knowledge-based systems, focussing on acquisition of knowledge, hardware/software, and transfer and export of the systems. The panelists were: Harry Pople (CMU), Bruce Buchanan (Stanford), John S. Brown (BBN), and Raj Reddy (CMU).

Harry Pople was the first panelist to express his concerns.

The expert's knowledge needs to be conceptualized in the proper way in the design of a knowledge-based system. Invariably, initial conceptualizations will be inadequate and re-conceptualization will be necessary. Therefore, the system designer should not become too emotionally committed to the code but should treat it as "throw away" code to be used for experimentation. The task is basically an empirical one and the ever present question is "Do the observations support the theory?". In this case the program is the theory and the expert's problem solving behavior and conclusions are the observations. Because this is an empirical task, it is important to get quickly into the experimentation phase and to interact with the expert.

Bruce Buchanan listed some of his observations in building knowledge-based systems.

1) The choice of expert is absolutely critical. It is important to have an expert who is sensitive to the computer science aspects of the task.

2) It is necessary to make some compromises in working with the experts. Experts want to see immediate feedback and immediate results. They are not concerned with developing AI techniques in the program.

3) The choice of conceptual framework is important. In MOLGEN for example, there are many levels of conceptualization and representation of the data.

4) The initial formalization of the knowledge will be different from the final formalization. It is important to plan for modification of the rules.
5) The program rather than a person should be able to take charge of "funnelling" knowledge from a number of experts into the program.

6) The explanation of the program's reasoning strategies that is given to the expert is important to the acquisition and debugging of the knowledge base. There is a need to distinguish between explanation and justification.

7) It is necessary to have at least the illusion of English in the interactions.

8) The character of the program can change dramatically over time, and the designer should keep this in mind when planning the system. MYCIN for example was initially a consultation system, but now it is being used for teaching as well.

John S. Brown described some research concerns at BBN.

Systems are being designed at BBN for naive users such as students in military training programs. The emphasis in these systems is on human engineering. To prove the validity of the systems it is necessary to have efficient systems with guaranteed response. It is possible to write efficient programs in INTERLISP, but working set size and associated page faulting are problems. LISP machines, which should be available in a few years for $50,000 - $60,000 are a partial solution since they will make it possible to guarantee response time. There will be problems however, with introducing personal machines into the current environment of large computing utilities.

In addition to efficient hardware, new inferencing mechanisms are needed. In CAT for example, the student must be given partial answers so that there is not a long delay between his response and the program's final answer.

Raj Reddy reported on the research at CMU which aims at reducing the cost of data acquisition.

Studies show that 50-70 per cent of a computing budget is used for data acquisition. Thus, it is desirable to have as a goal, the reduction of the cost of information gathering and data acquisition. Using speech is one way to do this, but it is necessary to have machine performance that is comparable to human performance, since people are intolerant of inferior machine performance in tasks in which they themselves are competent. Therefore, a goal for speech understanding
is real time performance costing 1 cent per second of speech. The problem now is to find out what needs to be done to get the cost this low.

Because current AI systems are slow is no reason to assume that they will be slow in the future. In fact, Newell and Simon show in "Multiplicative Speed-up in Systems", that speed-ups of one million or one billion are possible. This is because AI systems consist of a multiplicity of levels - technology, hardware/software...... knowledge sources, algorithms, each level having potential for speed-up. This means that if two levels are speeded up by a factor of 10, the overall speed-up is 100. Furthermore, is is not unreasonable to expect speed-ups of 10 or 100 at each level. Silicon technology for example, already has resulted in speed-ups of 100.

AI is not concerned merely with the study of intelligence, but also with the creation of intelligent artifacts. In this respect AI can be called "advanced computer science", as it acts as a "forcing function" on the entire field.

Two aspects of AI as advanced computer science are being investigated at CMU: (1) The optimization of representations and (2) problem oriented computer architecture design.

The problem of optimizing representations occurs in the context of the multiple, independent, co-operating knowledge sources of HEARSAY II. Computing time is spent on two tasks, deciding what to do, and doing it. The problem is to choose a representation which minimizes the first task and maximizes the time spent on the second. The solution has been to compile the knowledge sources. This means that the problem solving knowledge for speech recognition is retained, but the knowledge used to decide on the problem solving strategy is omitted.

Also being developed at CMU is computer architecture oriented to the problem of speech. A pipeline multiprocessor system is being designed which should achieve the goal of speech understanding in less than real time.

Tom Rindfleisch commented on exporting programs.

The practical use of knowledge-based systems has to be demonstrated. This means that programs must be exported. Programs have to be engineered properly in order to make them efficient and easily exportable. Currently, the only way to export programs is to recode them; no other method seems feasible at this time. Even the arrival of LISP machines will not eliminate the need to recode programs for export.
Before inviting discussion from the audience, Ed Feigenbaum expressed his views on some important issues.

A major problem in the AI field is the chronic lack of accumulation of knowledge. The knowledge, tools and expertise of the master builders are not being passed on to others. Confusion arises in the multiple and conflicting use of terms. For example, "frame", "template" and "schema" refer to the same concept, and there are a number of different concepts referred to as "frames".

There are several ways to alleviate the problem. One way is the AI Handbook, which is a collection of overviews, conceptualizations, techniques and example systems, and a joint effort by the community to put the knowledge into a polished, usable form. A second and traditional way of AI, is to incorporate new techniques into AI languages. A third way currently being tried at Stanford is the development of the AGE (Attempt to GEneralize) system. This system consists of a collection of software packages and tools with clear interfaces to knowledge bases. The generality is not in the nature of a method, but in having a collection of tools which, taken together, have general applicability. An example is EMYCIN, the MYCIN system with the knowledge of bacteria taken cut. Using this system, it was possible to develop a working pulmonary function program of 50 rules in about 50 man weeks.

Feigenbaum listed other issues for consideration.

1) The ultimate importance of the knowledge engineer's work may have nothing to do with developing a particular program. Rather, the contribution may be to the domain itself, in formalizing and structuring the knowledge of the domain, and in providing methods for doing so.

2) Knowledge-based systems take a long time to build. Speech projects received five year funding; DENDRAL was in existence eight years before it had significant use. Because of this long development time, funding agencies and reviewers must take a long range view of the tasks and the prospects.

3) In addition to a chronic lack of hardware, there is a shortage of knowledge engineers.

The ensuing discussion focussed on themes which developed out of the preceding comments of the panelists.

Knowledge Engineers
Joshua Lederberg responded to Feigenbaum’s comment that there is a shortage of knowledge engineers. He pointed out that researchers in other disciplines, linguists, librarians, researchers in mathematical programming and combinatorial problems share similar concerns with AI knowledge engineers.

Bruce McCormick agreed with Saul Amarel that the term "engineering" itself is objectionable in this context because the task is more scientific in character than the term implies. That is to say, the task is one of defining problems and setting boundaries. The way an AI system selects knowledge in problem solving is fundamentally no different from the way a scientist selects knowledge in any other scientific enterprise.

The Expert and Knowledge Acquisition

Lederberg stated that knowledge engineers perturb the knowledge as they enter it into the knowledge base.

In the course of collaboration, the expert undergoes a similar change. The expert becomes biased as he receives the knowledge engineer's interpretations as feedback. People used the phrase "perturbation of the expert" to describe this phenomenon.

Don Waterman said that in the initial interactions the experts themselves are not certain how they have formalized their knowledge or how they make decisions. These initial interactions therefore, are not useful to the knowledge acquisition process.

Raj Reddy disagreed. He said that all interactions with experts, even the initial ones are part of the knowledge acquisition process and cannot be ignored.

Suppression of Detail and Knowledge Compiling

Lederberg stated that in the practice of knowledge engineering much detail is left cut as the knowledge is distilled to fit the requirements of a particular program or formalism. For example, in INTERNIST, the numbers on the links are a compact representation of considerable medical knowledge. It is not wise to forget the suppressed detail since it will be needed to make changes and additions to the knowledge base.

Raj Reddy described the problems that suppressing detail causes in speech understanding. He cited the "sit/slit" problem as an example. These two words sound alike and a set of recognition rules produced by a knowledge compiler might not be able to make the distinction. It is
necessary then, to go back to the omitted detail that was used in creating the rules, in order to perform the recognition successfully.

Peter Szolovits pointed out that suppression of detail occurs at the level of interaction between the knowledge engineer and the domain expert as well, and that only a fraction of their interaction ends up in the program. Therefore, it is premature to expect the machine to do the knowledge distillation.

Lederberg said that sometimes experts will consciously ignore certain details because they consider them unnecessary at that particular stage of the system's development. At other times this suppression may be less conscious, as when the detail deals with the underlying basis and justification for the inferences themselves.

Pople felt that the experts perform a kind of reasoning that so far has not been captured in programs. Their patterns of reasoning are too subtle. Some of the diagnostic reasoning of Dr. Myers is in the INTERNIIST program, however the program is unable to refer to basic principles of anatomy or physiology for example, that Dr. Myers can refer to when the need arises.

Buchanan agreed. Humans have a way of using the interplay of several knowledge representations and we have very little understanding of the way in which this is done. Programs however, only have one way of looking at the problem, and the user must be made aware of this.

Dr. Myers added that the INTERNIIST data base can be regarded as a descriptive text book of medicine. In some instances information is hidden. In other instances the knowledge is simply too complex to represent.

Amarel suggested that rather than suppressing detail, the knowledge engineer is choosing an appropriate grain of representation. The choice of the grain depends on the particular task. "Knowledge for what?" is the important question, and it is a difficult one to answer because of the ever present problem of having to define the boundaries of the task. This makes it difficult to know exactly what knowledge is needed. It requires exploration and experimentation. In current systems, the knowledge was put in for a particular task and if the system is to do more, the knowledge must be enriched.

Raj Reddy added that knowledge compilers can compile a knowledge source into a form which is specialized for a particular application.

Randy Davis thought the idea of a knowledge compiler is beneficial because it makes the program efficient. On the other hand, it is clear that one cannot work solely with compiled knowledge.
Bill Martin argued that there is nothing new in these ideas about compiling knowledge. They are old ideas and techniques being applied to problems of a larger scale.

Amarel disagreed. He said there are new ideas here which have not received serious attention. The notion of a knowledge compiler which would transform knowledge from one representation to another automatically, is unlike GPS or the heuristic compiler.

**Multiple Representations and Multiple Purposes**

Raj Reddy stated that the way in which knowledge is used may require different representations, for example, one representation for inferencing and another for explanation. Explanation seems to require much of the suppressed knowledge.

Buchanan thought that ultimately, it will be necessary to use one integrated database for multiple purposes. Such important considerations as consistency are facilitated by having a single source of knowledge. Furthermore, it is difficult to decide after re-representation whether or not one still has the same knowledge.

The assumption that there is a single knowledge source is not justified, Sridharan argued. In the CASNET/GLAUCOMA application there are two very different models: a model using a system of parameters and differential equations, and the causal model. It is not clear at all how to move from the differential equation model to the causal model and it is unreasonable to expect a program to do it. In reality these are not transformations of one source but two separate and different sources.

Randy Davis suggested that perhaps these two different representations in the CASNET/GLAUCOMA system are already abstracted from one root source.
The panel comprised members of the AIM Advisory Committee. These were Joshua Lederberg, Ed Feigenbaum, Don Lindberg, Jack Myers, Dan Bobrow, Aran Safir and Saul Amaral.

Discussion focussed on what direction the resource should take to insure future growth and funding.

Bruce McCormick argued in favor of a strong centralized resource which would serve as a national laboratory, able to accommodate even the large information processing industries like health care delivery. To bring the SUMEX resource to the level of a national facility would require a strong community backed by sympathetic congressional leaders, who would go directly to Congress for funding.

McCormick stated that medical problems are an international concern, and there are researchers engaged in parallel activities in Europe for example, who would like to share similar facilities. The SUMEX community should do something to establish collaboration with these researchers. The competition would be healthy and the community is losing much of the world's talent by not doing so.

Both Lederberg and Feigenbaum argued against these views. Their objections were based on the obstacles they saw in the way of this kind of immediate large scale growth.

Feigenbaum pointed out that AIM has not produced a single product for use by anyone outside the community. The research does not have the credibility as yet to win large scale support from Congress. There are many obstacles in the way of building that kind of complexity - obstacles not only on the hardware level but on the administrative level as well.

The alternative is to support the development of similar projects and to work in communication with them in long range planning so that when it comes time for further support, the funding agencies will see coherence among the resources. The AIM community cannot expect to be the "unique voice" for computing in medicine - it is enough that the resource meets the current users' requirements.

Regarding international collaboration, Lederberg agreed that the intellectual traffic that it would bring about is attractive. However, the resources needed to make this possible have never been
adequate in other countries. There are still too many hardware and expert problems to overcome.

The discussion included suggestions for improving the dissemination of technical information over SUMEX. The need to use the system's mail service more for this purpose was stressed. Potrow suggested there be a way to notify users about current papers and to make them available over the net. This would help the community to maintain contact throughout the year rather than to wait for an annual workshop to share research problems and developments. Lederberg referred to the SUMEX BULLETIN BOARD as such a facility. However the bulletin board is an intra-SUMEX facility and there has been no thought of making it public. The possibility of having an electronic journal was suggested, but this met with criticism. It would require editors and staff which is costly; quality would diminish if articles were entered without being refereed first, and there are enough printed journals of excellent quality available already.

Another suggestion was that SUMEX could have a command called "EXPERT" which would accept problem topics and would respond with a list of experts who could help solve the particular problem immediately over the network. Lederberg's opinion was that SUMEX already serves as a query distributor, and that experts do not want to have their names made available in that way. The alternative is to have the system channel queries to the appropriate expert and to wait for a response. All users face the same frustrations including the users at Stanford. The experts at SUMEX are not always available to answer questions. Their time is limited and often they have jobs to do at SUMEX which are more important than servicing a user's immediate needs.

Lederberg went on to express some concerns of the AIM Advisory Committee.

The question of choosing the best means for recruiting people and resources is important to the growth of the community. There is no better means to advertise for the facility than through its users. Elliott Levinthal is the official liaison at SUMEX for recruitment.

The committee is also concerned that there may be programs that could be mounted and used by the community but are not being made available because the system is overcrowded already.

The experience of the resource is important to others who wish to establish similar projects. How other fields benefit from SUMEX in terms of services and resources is of major concern to the AIM Advisory Committee.

The whole question of how such facilities will be handled in the future is uncertain, since a single indiscretion on the part of a
user could jeopardize future funding. Although it is difficult to see the details along the path to five years from now, Lederberg foresees many small systems in use at that time. Community organization, plus some sharing facilities to keep track of additional support will be essential. The issue of community will be ever more important. Generous support for increased distributing capability will be needed to avoid the "Tower of Babel" that could arise if communication among resources is not maintained.

SUMEX is an excellent model of a national resource. It has been successful enough to gain additional funding, and this will be used in part to increase the memory capacity of the system, thereby providing 30 per cent more throughput capability.

The resource is built around a community of researchers and is attuned to the needs of that community. As such, it not only provides the facilities for research but it helps to direct the course of the research as well.

The AIM community is most fortunate to have Bill Baker as its liaison at NIH. He has been a powerful force behind this effort and has been most influential in securing support for the project.