

Time-Dependent Features

A consultation system built under the current design of EMYCIN takes a snapshot of the available information about a case and makes a one-time evaluation of the situation. In cases where the nature of the diagnosis or repair is strongly dependent on an understanding of the process of failure over time, this static approach to the problem is inadequate. No provision is made in the present system for considering the same case several days later when more information is available or when the values of some parameters have changed.

The system also lacks a mechanism for dealing with parameters whose values vary with time. In many domains, time considerations may be crucial to the solution of even the simplest problem. For example, it might be critical to track the values of various parameters over a period of time, or to check what value existed at a particular time in the past.

In order to increase the number of domains in which EMYCIN systems will be useful, we plan to add two new features. The first is a "restart" mechanism that will allow a user to run a follow-up consultation on a stored case, adding information that has become available since the original consultation, and correcting old answers that are no longer accurate. The second is to expand the syntax and semantics of rules to deal with values of parameters changing over time.

Follow-up Consultations

The builder of an EMYCIN system should be able to specify which parameters are likely to change for a given case from one consultation to the next. In a follow-up consultation, the system should summarize its knowledge of the case and do the following three things:

- 1) ask whether new information is available for any of the parameters which are subject to change, and prompt for the new answers;

- 2) ask whether values are known for any of the parameters whose values were UNKNOWN at the time of the previous consultation, and prompt for the new answers;

3) allow the user to specify changes which may have occurred in the values of any other parameters (viz., those which do not usually change).

Extending the Rule Syntax and Semantics to Deal with Time Relations

The builder of an EMYCIN system should be asked to classify parameters according to their stability over time. A possible classification scheme is shown below.

- 1) Constant - value is always the same (e.g., Name and Sex of medical patients)
- 2) Regularly changing - new value is available at regular intervals; there will be several values stored for the parameter, each with a time (e.g., barometric pressure at a certain city)
- 3) Gradually refined - value is likely to change over time, from unknown to uncertain to definite (e.g., Identity of an organism growing on a culture plate)

Parameters of the first type are the typical case that EMYCIN now handles. For the second type, a time must be kept with each value-CF pair. The third type of parameter will typically change from one consultation to the next, and previous values will be discarded as new information becomes available.

New PREMISE and ACTION functions must be defined so that EMYCIN rules can handle time-varying parameters. Functions will be needed to test and conclude (a) the value of a parameter at a given time, (b) the duration of a particular condition (e.g., it has been raining for three hours), and (c) trends in the values of numeric parameters (e.g., the volume of water in the tank has increased within the last hour). As we test EMYCIN in different domains, we may discover other types of tests and conclusions that must be made on time-dependent parameters.

Add Capabilities for Using Meta-Rules and other Meta-Level Knowledge

Our preliminary research with meta-level knowledge [15] as

well as our preliminary experience with the GUIDON tutorial program has shown the importance of acquiring, using and teaching structural and strategic meta-knowledge, as well as the domain rules. Structural meta-knowledge provides a framework that sets the context for domain rules, and in tutoring helps make the rules memorable to a student. It might include patterns and principles that are made specific by groups of rules. Strategic meta-knowledge constitutes planning knowledge for using the rules to solve different problems [10]. This meta-knowledge is written as meta-rules and takes the form of diagnostic reasoning strategies and domain-dependent approaches for efficient consideration of a case.

In our work with EMYCIN, we will explore various kinds of structural and strategic meta-knowledge that is appropriate to the production rule representation and useful for explaining decisions made by the program (to a consultation user or a student). We will start by implementing in EMYCIN the capabilities for using the meta-level knowledge described by Davis: meta-rules to be used for pruning and reordering the object-level rules, and meta-level models of rule sets that aid in debugging (and tutoring) the domain knowledge.

Experience with EMYCIN programs like HEADMED and PUFF will provide us with particularly useful case studies of possible forms of meta-knowledge.

Incorporating Question-Answering Facilities into the System

In order to make the questions-answering facility available to an EMYCIN consultation system, the system must be provided with a dictionary of synonyms and a list of definitions of the important concepts in the its domain of expertise. The dictionary will contain common synonyms in the domain, pointers between English words and parameters, and common phrases in the domain that can be given a single specified meaning.

We will provide a facility for automatically constructing a dictionary from the parameters in the knowledge base. The system builder will also be able to add synonyms and fill in parts of the dictionary that cannot be created automatically. This should provide all the information necessary for answering standard questions about the consultation system. The kinds of questions that the system will be able to answer are:

- 1) the value of a parameter

- 2) how a parameter was used or concluded in the consultation
- 3) how a parameter is used or concluded in general
- 4) how a rule was used in the consultation
- 5) why a question was asked during the consultation
- 6) the translation (into English) of a rule
- 7) the definition of a concept

These question types will be recognized in a variety of forms. For example, all of the following will be taken to be equivalent ways of asking for the value of a parameter

- 1) What is the value of X?
- 2) Is Y the value of X?
- 3) What is X?
- 4) Do you know what X is?

The major benefits of providing these capabilities are that the user of a consultation system can understand the reasoning and the designer of the system can find the sources of reasoning errors.

Coupling a Tutorial System to EMYCIN

Work on the idea of automatic "Transfer of Expertise" from a human expert to a program [22], [15] has led to important advances in the representation of knowledge within the program. These advances have allowed the systems to explain their reasoning process to users, thus providing the basis for a tutorial program. We have been building an intelligent computer aided instruction (ICAI) program [12] that guides a subject through problems in a complex domain with the goal of transferring the system's knowledge of the domain to the student.

Current ICAI techniques like planning the discourse, modelling the student, and teaching problem solving strategies all take a natural form in our system. In turn, the system serves as an excellent environment for experimenting with unsolved problems in the design of computer-based tutoring.

We have demonstrated the feasibility of using the MYCIN knowledge base for teaching as well as for consultation, and this aspect of our research will be continuing during the grant period under separate funding⁴.

We have not yet demonstrated the generality of the tutorial program, GUIDON, in other domains; but we have meticulously avoided introducing any domain-specific knowledge into GUIDON's control structure and teaching strategies. We believe that its design is as general as MYCIN's. Thus, all that is needed for tutoring in another domain will be (a) domain rules for EMYCIN to use on cases which GUIDON can discuss and (b) domain specific meta-level knowledge that would be useful for teaching these rules. Moreover, we must keep the tutoring strategies of GUIDON coupled to the representation of EMYCIN systems that we wish to tutor.

III.C.2. AGE-1

The basic idea behind AGE-1 is to generalize the ideas found in specific problem-solving systems and make them available in a package — hence the name AGE, for "Attempt to Generalize". AGE-1 takes an active role in assisting a knowledge engineer in constructing a performance system. The specific model that is incorporated in AGE-1 — the "cooperating knowledge sources model" — was pioneered in the HEARSAYII system ([20], [33]) for speech understanding. It was further developed by Stanford researchers in two data interpretation problems — SU/X and SU/P (otherwise known as HASP and CRYALIS) [43].

III.C.2.a. Examples from AGE-1

The CRYALIS program [19] is a knowledge-based program being developed in collaboration with the University of California at San Diego. Its task is to infer protein structure from X-ray crystallography data. This program was developed in

⁴Joint proposal to Office of Naval Research, Personnel and Training Division and Advanced Research Projects Agency.

close collaboration with the AGE group at Stanford and has been using a very similar problem-solving model. Currently the top-level of CRYNALIS is being rewritten using the AGE-1 package. Examples from the CRYNALIS program are used below to illustrate the problem-solving model in AGE-1.

The Problem-Solving Model

AGE-1 uses a uniform multi-level data structure, termed the "blackboard", to hold the status of the system. In CRYNALIS, the blackboard is used to hold various crystallographic data and structural hypotheses. Separate hierarchically organized panels of the blackboard correspond to "electron-density" space and "protein-model" space. These correspond roughly to data space and hypothesis space except that the electron density space has two levels of hypotheses above the electron density data. The protein-model space describes the three-dimensional structure of the protein at different levels of abstraction from the atomic level to the large-scale structural features like "beta-sheets".

Skeletal Level (backbone — graph of density nodes)	Stereotypic Level (helices, beta-sheets)
Nodal Level (high intensity points)	Superatomic Level (Side chains, proline)
Parametric Level (electron density data)	Atomic Level (C,N,Fe etc.)

Electron Density Space

Protein Model Space

A set of procedures termed knowledge sources (KSs) are used to form and link the hypotheses on these panels. In the CRYNALIS application, these knowledge sources include such domain specific operations as skeletonization, helix identification, sidechain identification, bond rotation, sequence identification, cofactor identification, and heavy atom identification. The knowledge sources are expressed as production rules. AGE-1 provides a framework for coordinating the activity of the KSs mixing goal-driven and data-driven reasoning as it searches for solutions.

If the KSs had been perfect, the coordination could have be

directed in a goal-driven manner analogous to the production rules in EMYCIN. However, because of gaps in the theory and implementation of the individual KSs and noise in the data, they are individually incomplete and errorful. Like the HEARSAYII system, AGE-1 uses an algorithm — a version of the hypothesize and test paradigm — which emphasizes cooperation (to help with incompleteness) and cross-checking (to help with errorfulness). During the hypothesize part of the cycle, a KS can add a hypothesis to the blackboard; during the test part of the cycle, a KS can change the rating of a hypothesis in the blackboard. This process terminates when a consistent hypothesis is generated satisfying the requirements of the overall solution or when knowledge is exhausted.

In AGE-1, the hypothesize-and-test paradigm is formalized as a control structure with three levels. The first level is the hypothesis-formation level. KSs on this level make changes to the blackboard panels. In the hypothesize and test paradigm, they put hypotheses on the blackboard and test the hypotheses of other KSs. A rating is associated with each hypothesis to store the overall judgment. Immediately above the hypothesis-formation level is the KS-activation level which contains two KSs. The KSs are called the "event-driver" and the "expectation-driver" and correspond to data-driven and goal-driven policies for activating KSs on the first level. The highest level of KSs is called the strategy level. This level must decide (1) how close the system is to a solution, (2) how well the KSs on the second level are performing and (3) when and where to redirect the focus-of-attention in the data space. KSs on this level can invoke KSs on the second level.

This problem-solving method is more complex and more general than the backward-chaining approach used in EMYCIN. It is designed to tolerate errorfulness in the data and in the KSs and allows the inferences to be run opportunistically in either direction. It also allows the inferences to be run at several levels of abstraction.

Using AGE-1 to Build a Knowledge-based System

The purpose of the AGE-1 system is to assist a computer scientist at building a problem-solving system. AGE-1 is intended to speed up process task when the task domain can be cast in the model of cooperating knowledge sources. To this end, AGE-1 has several software subsystems — a "TUTOR" subsystem and several knowledge acquisition subsystems.

The TUTOR is a module for the unfamiliar user which helps

him create an application program. It guides the user through a top-down design of his system by presenting him with a list of topics and subtopics at each level. Canned text is available for explaining the choices at each level. A "browse" option is available for random perusal of the topics and subtopics.

Knowledge about the parameters of the application program is acquired by the DESIGN subsystem. The DESIGN subsystem provides the user with choices at each phase of the construction of the application program. This construction involves choices for hypothesis structure, rule acquisition, goals, and expectations. Thus, the domain dependent particulars for each of the components of the application program are asked about in turn. For example, the following items must be acquired for each KS

1. preconditions
2. inference levels
3. links
4. hit strategy
5. local variable bindings

The acquisition of each of these items is further broken into the most primitive elements. The DESIGN module has a "guided" approach for the novice and an "unguided" approach in which an expert calls for the knowledge acquisition functions quickly and directly.

III.C.2.b. Applications of AGE-1

The CRYSLIS example illustrates the most comprehensive application of AGE-1. AGE-1 has also been used on an experimental basis to create a version of PUFF Section III.C.1.b. and on some cryptography problems (simple code-breaking). These applications have been used for testing the tutorial and knowledge acquisition components of AGE-1.

III.C.2.c. Proposed Work for AGE-1

In the current version of AGE-1, the DESIGN module provides choices and explains them with canned text. AGE-1 does not build up its own knowledge of the user's application — only a knowledge of the design choices that the user makes. It does not make inferences about the relationships between design choices — so that it does not infer choices for the user even when one set of choices implies another set.

We plan to move toward a system where AGE-1 will ask the user about the domain and play a more active role in making the design decisions. This means that AGE-1 must have a model of "how to build a system" and that we must encapsulate the reasons behind the design choices. Our plan is to begin to capture this information in the form of production rules which relate the form of the domain knowledge to the design choices of AGE-1 to a prediction of the performance consequences in the application program being built.

Accompanying this effort we would like to begin construction of two explanation subsystems — one for explaining the activity in the design phase and one for explaining performance of the application system. We expect to build on the explanation work in the EMYCIN system for this.

In the long term, we also plan some work on knowledge compiling. Our plans for this in the EMYCIN system have already been discussed. There is some experience in compiling the knowledge of a cooperating knowledge source system — notably the HARPY [39] system which can be seen as a "compiled" approach to the task performed by HEARSAYII. Much more work is needed before this could be done automatically.

III.C.3. The Unit Package

The Unit Package is a frame-structured representation system developed as a tool for building knowledge bases in the MOLGEN project. Unlike EMYCIN and AGE-1, the Unit Package provides no problem-solving framework. However, the Unit Package can be used as a passive representational medium in conjunction with specific problem-solving approaches. Two approaches to experiment planning are being developed in this way as part of research in the MOLGEN project. The Unit Package is also accessible from within the AGE-1 package. The Unit Package builds on a substantial amount of work (both here and elsewhere)

on frame-structured languages. A comprehensive description of this work is available as a technical report [52] which is included with this proposal.

Knowledge in the Unit Package is organized in a semantic network of nodes and links. Following other work on frames [42], the nodes are called "units" [6] and the links are called slots. The major software components of the Unit Package are (1) an interactive editor for adding new information or modifying existing information, (2) a set of routines for matching and manipulating descriptions, and (3) a set of access functions which maintain network relations (such as inheritance of properties) and provide an extended address space to hold the semantic network.

III.C.3.a. Examples from the Units Package

The Unit Package is a fairly extensive set of software for defining the symbolic entities of a domain. It provides a number of conventions and methods for defining standard kinds of relationships between the symbols.

There are three main steps building a knowledge base for a domain with the Unit Package. The typical user of the Unit Package is a computer scientist, although four geneticists on the MOLGEN project routinely use the Unit Package. The main steps are using the interactive editor are as follows.

- (1) Define the symbols of the domain. These symbols take the form of units as illustrated below.
- (2) Define the operations which manipulate these symbols. Operations are procedural knowledge in the form of production rules or LISP functions.
- (3) Define an approach for problem solving.

The steps are not necessarily performed in this order or by one person. In an evolving knowledge base, the user uses the editor both to create new symbols and to modify old ones as his understanding improves. The expertise to define all of these things may be spread over several people working on a common knowledge base.

"Specialization" is a relation which is indicated by a user when he defines a symbol. It is used to indicate subclasses among concepts — e.g., the unit for the restriction enzyme Eco RI is a specialization of the unit for general restriction enzymes which is a specialization of the unit for endonuclease which is a specialization for the unit for nuclease and so on. General properties of a class are inherited by its specializations. This is formalized in part by having descriptions in slots of those units that correspond to classes. These descriptions delineate legal values for the corresponding slots in specializations of the class. Descriptions can be progressively tightened as one proceeds down a specialization hierarchy. This feature makes the process of specialization correspond to the addition of non-contradictory new knowledge to units. A specialization (or generalization) hierarchy of concepts from a molecular genetics knowledge base is illustrated below.

LAB-OBJECT

```

ANTIBIOTIC
  AMINOGLYCOSIDE
    KANAMYCIN
    NEOMYCIN
  BETA-LACTAM
    AMPICILLIN
    ...
GENE
  APR
  CMR
  ...
ENZYME
  LIGASE
  ...
  NUCLEASE
    ENDONUCLEASE
      RESTRICTION-ENZYME
        ALU1
        ASU1
        ...

```

Symbols in the Unit Package are organized in a generalization hierarchy. This hierarchy indicates "inheritance paths" by which symbols acquire the attributes of their generalizations.

Each of the symbols in a knowledge base is defined in terms of "slots". A unit corresponds approximately to a property list

except that (1) the structure of a slot has several explicit fields for information about such things as modes of inheritance and datatype and whether the value is stored or computed⁵ and (2) the value of a slot can be a description of a value. The following figure illustrates two units of different complexity.

NAME:	Endonuclease
DOCUMENTATION:	A nuclease that cuts internally in a DNA structure.
SITE-TYPE:	One of (MONO, STICKY-HEXA, FLUSH-HEXA, PENTA, STICKY-TETRA, FLUSH-TETRA)
3'-END:	One of (P, OH)
5'-END:	One of (P, OH)
MODE:	One of (Precessive, Non-precessive)
OPTIMAL-PH:	RANGE (Ø 14)
...	...

NAME:	Rat-Insulin-Problem
DOCUMENTATION:	This unit gives the parameters of an experiment for cloning the gene for rat-insulin.
GENE:	RAT-INSULIN
GENE-PRECURSOR:	RAT-INSULIN-RNA
ORGANISM:	A Bacterium
Default:	E.COLI
VECTOR:	A Vector
GOAL:	A Lab-goal with
	STATE = A Culture with
	ORGANISMS = A Bacterium with
	EXOSOMES = A Vector with
	HAS-GENES = RAT-INSULIN
	CONDS = (PURE? ORGANISMS)

Two units from a MOLGEN knowledge base. Each unit is organized as a list of slots. The slots are filled with values or descriptions of values. These units are examples of "symbols" from the molecular genetics domain.

While the Unit Package is not a problem-solving program, it does provide a large number of routines for creating, modifying, and matching units in a knowledge base. These routines are called by problem-solving programs in the MOLGEN project which are currently being tested. Some of the built-in features — such as the generalization hierarchy and symbolic descriptions — seem to be especially useful for problem-solvers that work with

⁵See the technical report for details.

abstractions. For a discussion of other features of the Unit Package — such as the various modes of inheritance, set notation, or the attachment of procedural knowledge — the reader is referred to the enclosed technical report.

III.C.3.b. Applications of the Units Package

MOLGEN — Planning Experiments in Molecular Genetics

Molecular genetics is a rich and rapidly growing science. Several aspects of molecular genetics make it attractive as a task domain for artificial intelligence. It is a young science and new techniques and ideas are developed regularly. This makes it attractive for studying the process of discovery ([38], [23]). It is a laboratory science and experiments are clearly defined in terms of laboratory steps and results. This makes it attractive for studying the processes of planning and plan debugging. Finally, many kinds of knowledge are used in molecular genetics. This motivates work on representation in the Unit Package.

Planning research in MOLGEN has focused on two broad classes of experiments — structural synthesis and structural analysis. The synthesis experiments use various laboratory techniques to build DNA structures. Analysis experiments use various laboratory techniques to identify an unknown structure. An analyst seeks to discriminate between competing hypotheses for the structure of a sample.

Other Applications

In the past few months, several other projects have begun to use the Unit Package as a representational medium. Dr. Blum [5] is using it in an application which will combine statistical methods and AI methods for performing studies on a clinical data bank at Stanford. The Unit Package is being used to represent a set of medical models to permit a more sophisticated interpretation of patient record data in the data base than is possible using statistical methods alone.

The Unit Package is also being used in a mathematical application at Stanford and is being tested for a planning application at the RAND corporation. Other applications are expected over the course of this grant period.

III.C.3.c. Proposed Work in the Units Package

The proposed work on the Unit Package may be divided into two main categories — representational work and research-related work. Barring surprises from the emerging applications of the Unit Package, most of the work on representational machinery is finished. There are a few outstanding tasks such as (1) generalizing the concept hierarchy to be a concept graph so that units can have more than one generalization and (2) providing some more flexible forms of inheritance. Since the Unit Package became operational in June 1977, the rate of change to the system itself has slowed dramatically. This reflects the need for a stable system for development of applications and the fact that the Unit Package has found an important niche for the applications in the Heuristic Programming Project.

This standstill in development also reflects the current interests of the research group — which is to work on the problem-solving applications of the Unit Package. A great deal more development will become important as this work is completed. For example, the Unit Package provides a substantially richer descriptive language for concepts than is available in MYCIN or EMYCIN. It lacks, however, substantial facilities for knowledge acquisition — beyond a simple interactive editor. As applications of the Unit Package develop, an increased need for a stronger user interface is expected — incorporating such things as the natural language interface (BAOBAB [8]).

Another line of development is the development of standard relationships which appear in many domains. The Unit Package currently provides only a very small set of built-in relationships — such as generalization and specialization — which are utilized by the semantic network processing functions. Creating additional relationships is part of the knowledge-engineering task of applying the Unit Package to a task domain. Some of these relationships — such as "part-of" or "abstraction-of" — seem to appear in many domains. To the extent that these relationships have general utility and can be standardized, they will be made part of the initial knowledge base for new applications — thus expanding the apparent power of the Unit Package and reducing the effort of starting new applications.

III.C.4. Long Term Work and New Packages

The development of packages over the next five years will be opportunistic — relying on the most usable results from core research in artificial intelligence. Thus, while the following

ideas indicate only our best current ideas for continued development.

III.C.4.a. Planning Package

One of the areas in which we see future work is in the general area of planning. The artificial intelligence research on this problem is currently being performed in the domain of experiment planning in molecular genetics. Some interesting ideas are just beginning to emerge from this work which, if successful, could become the basis of a "planning package".

This research is investigating the viability of a new approach to planning called "orthogonal planning". The thrust of this approach is to take the elements of a planning out of a "planning algorithm" and put them into explicit "planning spaces". Explicit planning operations such as refinement (mapping from abstract to specific) and evaluation and subgoal proposing are expressed as operators in a planning space. Different combinations of these operators can be arranged to create top-down (goal-driven) planning, bottom-up (opportunistic) planning, and various hybrid methods. The planning research seeks to find general methods for deciding when to apply these different planning operators in order to plan flexibly and effectively. Currently ten planning operations have been formalized in the planning space and four strategic operations have been formalized in a overseeing "strategy space". This approach is being tested in the domain of experiment planning in molecular genetics and uses the Unit Package for representing the symbols and operations in all of the spaces.

III.C.4.b. Time-Oriented Knowledge Representation Package

One important topic in computer-based diagnosis and therapy programs is the representation of knowledge about situations that are changing over time. Most current programs have concentrated on the interpretation of a single instance in the course of the patient's disease process. As the patient status changes over time, a program must be able to modify its representation to conform to the new situation. The ability to represent trends in the health of the patient is an important part of the diagnostic process.

Creation of a package that supports the representation of

changes over time will be important for applications based on clinical data bases. These data bases typically contain the results of a variety of tests which were administered at each patient visit to the clinic. The problem of interpretation of updated test results has also come up in each of our current applications, for example, initially negative culture results that grow out a particular pathogen after several days in our infectious disease program or the comparison of new pulmonary test results with the previous findings. No general purpose approach has been incorporated into these programs.

A program for a particular dynamic clinical setting - interpreting measurements from the intensive care unit has been developed at the Heuristic Programming Project. That program, named the Ventilator Manager (VM) [21], is able to evaluate a stream of thirty measurements provided on a 2 - 10 minute basis by a computer-based physiological monitoring system. The system: (1) provides a summary of the patient physiological status appropriate for the clinician; (2) recognizes untoward events in the patient/machine system and provides suggestions for corrective action; (3) suggests adjustments to ventilatory therapy based on long-term assessment of the patient status and therapeutic goals; (4) detects possible measurement errors; and, (5) maintains a set of patient specific expectations and goals for future evaluation.

Removing the the basic assumption about the regularity of the changes in the ICU setting is the major area of research in the development of this package. A typical problem is the interpretations of a series of test values that are higher than normal over several testing instances. Specialized knowledge about the typical rate of change of the underlying disease process is necessary to determine whether these values represent a trend.

The representation of dynamic settings also requires a model of the stages of the disease and treatment process that best characterize the clinical status of the patient. Often a particular value of a measurement takes on entirely different interpretations based on the current context. For example, the meaning of critical measurements one hour after surgery compared to the same measurement after three days of recovery. A rudimentary model of this type based on various therapeutic regimens is built into the ICU measurement interpretation system. Additional work is required in the generalization of this type of modeling process.

Codification and Use of Medical Knowledge
from Clinical Laboratories

ADMINISTRATIVE INFORMATION ONLY

1. TITLE OF PROPOSAL (Do not exceed 53 typewriter spaces)

Clinical Laboratory Exnert Project

2. PRINCIPAL INVESTIGATOR		3. DATES OF ENTIRE PROPOSED PROJECT PERIOD (This application)	
2A. NAME (Last, First, Initial) Lindberg, Donald A. B.		FROM [REDACTED]	THROUGH May 31, 1984
2B. TITLE OF POSITION Director, Information Science Group Director, Health Care Technology Center		4. TOTAL DIRECT COSTS REQUESTED FOR PERIOD IN ITEM [REDACTED]	5. DIRECT COSTS REQUESTED FOR FIRST 12-MONTH PERIOD
2C. MAILING ADDRESS (Street, City, State, Zip Code) University of Missouri 605 Lewis Hall Columbia, Mo. 65211		6. PERFORMANCE (See Instructions) [REDACTED] University of Missouri Columbia, Mo. 65211 Stanford University Stanford, California	
2D. DEGREE M. D.	2E. SOCIAL SECURITY NO. [REDACTED]		
2F. TELEPHONE DATA Area Code: 314 TELEPHONE NUMBER AND EXTENSION: 882-6966			
2G. DEPARTMENT, SERVICE, LABORATORY OR EQUIVALENT (See Instructions) Health Care Technology Center			
2H. MAJOR SUBDIVISION (See Instructions) Graduate School			
7. Research Involving Human Subjects (See Instructions) A. <input checked="" type="checkbox"/> NO B. <input type="checkbox"/> YES Approved: _____ Date _____ C. <input type="checkbox"/> YES - Pending Review		8. Inventions (Renewal Applicants Only - See Instructions) A. <input checked="" type="checkbox"/> NO B. <input type="checkbox"/> YES - Not previously reported C. <input type="checkbox"/> YES - Previously reported	
TO BE COMPLETED BY RESPONSIBLE ADMINISTRATIVE AUTHORITY (Items 8 through 13 and 15B)			
9. APPLICANT ORGANIZATION(S) (See Instructions) The Curators of the University of Missouri 215 University Hall Columbia, Mo. 65211		11. TYPE OF ORGANIZATION (Check applicable item) <input type="checkbox"/> FEDERAL <input checked="" type="checkbox"/> STATE <input type="checkbox"/> LOCAL <input type="checkbox"/> OTHER (Specify) University	
10. NAME, TITLE, AND TELEPHONE NUMBER OF OFFICIAL(S) SIGNING FOR APPLICANT ORGANIZATION(S) H. Kent Shelton Asst. Vice President Financial Services Telephone Number (s) 314-882-3512		12. NAME, TITLE, ADDRESS, AND TELEPHONE NUMBER OF OFFICIAL IN BUSINESS OFFICE WHO SHOULD ALSO BE NOTIFIED IF AN AWARD IS MADE H. Kent Shelton Asst. Vice President Financial Services 215 University Hall Columbia, Mo. 65211 Telephone Number 314-882-3512	
		13. IDENTIFY ORGANIZATIONAL COMPONENT TO RECEIVE CREDIT FOR INSTITUTIONAL GRANT PURPOSES (See Instructions) Graduate School	
		14. ENTITY NUMBER (Formerly PHS Account Number) 43-6003859	

PROJECT 3. Codification and Use of Medical Knowledge from Clinical Laboratories

ADMINISTRATIVE INFORMATION ONLY

RESEARCH OBJECTIVES

NAME AND ADDRESS OF APPLICANT ORGANIZATION

University of Missouri-Columbia

NAME, SOCIAL SECURITY NUMBER, OFFICIAL TITLE, AND DEPARTMENT OF ALL PROFESSIONAL PERSONNEL ENGAGED ON PROJECT, BEGINNING WITH PRINCIPAL INVESTIGATOR

Donald A. B. Lindberg, M.D., [REDACTED], Director, Health Care Technology Center and Information Science Group; Professor of Pathology
 Robert Abercrombie, Ph.D., [REDACTED], Post Doctoral Fellow, Information Science Group
 Paul Blackwell, Ph.D., [REDACTED], Professor of Computer Science
 Lamont Gaston, M.D., [REDACTED], Professor of Pathology
 Lawrence Kingsland, [REDACTED], Senior Electronics Technician, Information Science Group
 W. B. Stewart, M.D., [REDACTED], Professor of Pathology, Director of Laboratories
 Henry Taylor, M.D., [REDACTED], Professor of Pathology
 John Townsend, M.D., [REDACTED], Professor and Chairman, Department of Pathology

TITLE OF PROJECT John Yio Ph.D., 227 68 0029, Post Doctoral Fellow, Information Science Group
 Clinical Laboratory Expert Project [REDACTED]

USE THIS SPACE TO ABSTRACT YOUR PROPOSED RESEARCH. OUTLINE OBJECTIVES AND METHODS. UNDERSCORE THE KEY WORDS (NOT TO EXCEED 10) IN YOUR ABSTRACT.

A. Objectives

1. To represent within a computer-based information system the knowledge and procedures of the clinical laboratory expert.
2. To determine how to implement this information system such that benefits result to the clinical laboratory service which are measurable in terms of:
 - a. Increased quality of laboratory determinations
 - b. Reduced costs to the laboratory and/or the institution
 - c. Increased access to pertinent information by laboratory data providers and users.
3. To determine how to interface this information system with the hospital and clinic services such that benefits result in actual patient care. We propose to seek "process" measures rather than "outcome" measures.
4. Using this operational testbed to shed light upon certain important questions basic to artificial intelligence in medicine research.

These objectives will be pursued by construction of a knowledge representation system for the domain of the clinical laboratory expert. Subject matter expertise will be provided by directors of the clinical laboratories of the University of Missouri Medical Center. Fundamental artificial intelligence methodology and specialized computational facilities will be provided by the SUMEX Laboratory and the Department of Computer Science at Stanford University. Management and interfacing of the project and site-testing will be provided by the Health Care Technology Center at the University of Missouri-Columbia.

PROJECT 3: The Clinical Laboratory Expert Project

III. A. Objectives

1. To represent within a computer-based information system the knowledge and procedures of the clinical laboratory expert.
2. To determine how to implement this information system such that benefits result to the clinical laboratory service which are measurable in terms of:
 - (a) Increased quality of laboratory determinations
 - (b) Reduced costs to the laboratory and/or the institution
 - (c) Increased access to pertinent information by laboratory data providers and users.
3. To determine how to interface this information system with the hospital and clinic services such that benefits result in actual patient care. We propose to seek "process" measures rather than "outcome" measures.
4. To seek through this operational testbed to shed light upon certain important questions basic to artificial intelligence in medicine research. These include the following:
 - (a) How best to retain the power of symbolic representations traditional to AI techniques while at the same time obtaining the benefits of the numerical methods which are traditional to fields such as laboratory management?
 - (b) How best to set up an information system so as to accommodate to the endless stream of changes which occur in the operating environment of a system such as the clinical laboratory?
 - (c) How to improve, and hopefully optimize, the interface

of the knowledge engineer and the subject matter expert, in this case the clinical laboratory expert?

III.B. Background and Rational

Use of artificial intelligence techniques, especially the recent focus on formal representation of the knowledge of experts, is the latest and most promising of applications of the computer to medicine. It is already clear that the techniques are powerful and that the proof-of-concept and feasibility phases of medical applications have been successfully passed. This technique has been shown feasible in the areas of infectious disease (Shortliffe et al., 1973), glaucoma management (Weiss, Kulikowski, Safir, 1978), patient present illness (Pauker, Gorry, Kassirer, Schwartz, 1976), and in the general differential diagnosis in internal medicine (Lawrence, 1978). In many ways the AI techniques are still in development, but the real question remains: in what areas of medicine are they most usefully going to be employed? Some raise the question, in which areas would such techniques even be accepted?

The clinical laboratories offer the very best application sites for exploring AI techniques as a basis for biomedical information systems. The following observations support this contention:

1. The clinical laboratories were the first sites for successful implementation of computer-based information systems of any kind (Hicks, 1969; Lindberg, 1965, O'Kane, Haluska, 1977).
2. There are a host of current computer systems already disseminated in this field which form a basis for advanced technological developments.

3. Clinical laboratory services constitute a major part of hospital expenses (estimates vary from 25-40%).
4. Clinical laboratories, for the most part, are administered by professional medical personnel who have training in technological matters, including hardware and information systems, and who therefore are likely to be receptive to advances in this kind of methodology.
5. There is an expertise in clinical laboratory operation and interpretation which is recognized by medical specialty training.
6. Knowledge in this field is plentiful; and expertise takes the form of a multitude of tiny empirical pieces of information, which await unification into an overall information framework. This situation is compatible with the way in which formal knowledge systems have been built for other AI applications.
7. On the other hand, the field does offer an advantage in another (almost counter) sense: namely, that there are true and realistic models of the basic data generating sources. For example, one knows quite surely that impedance transients in a Coulter Counter are caused by particles, and that these particles are (for the most part) erythrocytes. Likewise, the concept of "serum electrolytes" is known to have a solid basis: namely, that there are actual, immutable ions of sodium, potassium, chloride, and bicarbonate (and CO_2) within the serum. Furthermore, chemical laws describe the relationship between many blood constituents. Curiously, the chemical laws are not used ordinarily as the

basis of laboratory management, and only partially as a basis for test interpretation and subsequent patient management. The chemical laws and the physical models are, however, a potential advantage in building advanced information systems.

8. The clinical laboratory offers a setting which is receptive to and safe for development of new information systems, yet which also offers a home base for extension out toward the more purely clinical setting. The meeting ground of the two is clear: it is the interpretation of the results of laboratory measurements.

For these reasons, we feel that clinical laboratories are in general a potentially fruitful place for AI in medicine applications.

There are reasons which make us think that the particular laboratories and group at the University of Missouri are a good choice among those institutions with excellent clinical laboratory programs.

1. The school has a long history in lab system developments. The first automated lab system in the country was built here in 1962 and has operated continuously since then.
2. The system incorporates all clinical laboratories and all test results.
3. These results are in computer processible form, indeed are reported through the computer systems. Consequently test data is accessible.
4. Experts in clinical laboratory medicine are members of the team who propose to build the Clinical Laboratory Expert system.
5. The project is sponsored by the Health Care Technology

Center, which has ample experience and capability in the management and conduct of multi-disciplinary technical projects. The Center management review of all projects includes participation of an evaluation team with members from operations research, medical sociology, economics, health services management, and medicine.

6. Most important of all, we have a plan to accomplish the system building, and we have predecessor systems to build on and to compare with.

III.C. Methods of Procedure

We propose to grow the information system beginning with a nucleus or model system and to expand the scope of the system by adding to it information and values from additional areas. That is, our strategy will be to begin with what is clearly feasible, to build our collaborative patterns about an early success, and then to expand in a systematic fashion to more ambitious goals. We feel this is not only a good general management strategy but the best way to build programming systems too.

Eventually, for instance it would be desirable for the system to be able to learn from the data. First, however, the system must be given the logic by which laboratory data are evaluated and understood. We plan for development of the system in four phases.

Phase One: incorporate the medical logic which takes into account the information which is available within the laboratory itself: e.g. test results, quality control results, methodological information.

Phase Two: Incorporate the additional medical logic which takes into account information about the patient; first simple aspects such as gender, age, race; then more complex concepts such as drug therapy,

operative status, clinical service assignment and provisional diagnosis.

Phase Three: incorporate medical logic which includes concerns for hospital function.

Phase Four: incorporate medical logic which attempts to link to considerations which are outside the hospital setting.

Following is a more detailed description of the phased development.

Phase I. The aspect of the lab results which is of primary concern within the laboratory hinges upon quality control considerations. These are the first logical aspects which must be represented.

We are referring initially to thinking which currently goes on strictly in the laboratory, previous to release of a test result. Subsequently, there may or may not be significant discussion between the laboratory director and the clinician concerning further lab work and/or clinical concerns. Previous to this stage, however, there is a great deal of evaluation done now within the lab and based on laboratory on only partially clinical grounds. Not enough evaluation of this sort is possible with today's high volume instruments. This function can be greatly enhanced by advanced computational techniques.

We would plan to introduce knowledge into the system along the following lines:

1. Knowledge of the labs selected (likely we would start with hematology and clinical chemistry)
2. Knowledge of what tests are done, what methods are used, what parameters are estimated, what units are used. It should be noted that there are often multiple extant methods for a single determination, as well as multiple laboratory locations throughout the institution at which it might be

done. Methodology and unitage change continually. Since a referral-type laboratory may do 3,000-5,000 different determinations, it is a serious problem to choose a representation which will be amenable to the endless updating

3. Knowledge of the kinds of patients and hospital locations.
4. Logic permitting an initial evaluation of the test result for credibility. This naturally includes arithmetic ranges, formats, etc.

5. Logic permitting evaluation taking into account other results from examinations performed as a battery.

An example is the well known relationship between hemoglobin and hematacrit.

6. Logic permitting evaluation of test result taking into account laboratory quality control procedures and records.

We have recently completed an evaluation of the proposed Bull statistic for control based on a weighted-moving-average of mean corpuscular hemoglobin concentration, which is a slight but still insufficient improvement on the traditional method.

This is an example of the need to bring numerical methods into alignment with the symbolic logic. In essence, this asks the general question, is it likely the result is valid considering the quality of the particular "run" or batch which produced the result?

The outcome of all the laboratory logic is the resolution of the following questions:

- a. Should the test be repeated using the same blood sample?
- b. Is the issue important enough (or specimen identification sufficiently questionable) that a new specimen must be obtained?