ARTIFICIAL INTELLIGENCE RESEARCH IN ANESTHESIA AND INTENSIVE CARE

*Glenn D. Rennels, MD, PhD,**
and Perry L. Miller, MD, PhD†

Rennels GD, Miller PL. Artificial intelligence research in anesthesia and intensive care.
J Clin Monit 1988;4:274–289

ABSTRACT. This article describes several research directions exploring the application of artificial intelligence techniques in anesthesia and intensive care. Artificial intelligence can be loosely defined as the discipline of designing computer systems that exhibit "intelligent" behavior. This article first introduces artificial intelligence and computer science research and discusses why medicine has proved to be a challenging domain for applying artificial intelligence techniques. A discussion of the central research themes that arise in medical artificial intelligence, many of which are common to different projects and to different medical settings, is followed by a description of specific research projects that apply artificial intelligence techniques in anesthesiology, ventilatory management, and cardiovascular management. Finally, further comments are made on the current state of the field.

KEY WORDS. Artificial intelligence. Anesthesia. Intensive care. Computers.

Computers and computer-related technology are increasingly evident in both the operating room and intensive care unit. It is clear that the use of computers in these areas will become increasingly sophisticated. The computer has the potential not only to gather, store, and display data, but also to play an active role in assisting the physician with the clinical decisions involved in patient care. A number of research projects are currently underway that will contribute to this expanded role for the computer in clinical medicine. This article gives an overview of several such projects, all exploring the application of ARTIFICIAL INTELLIGENCE (AI) techniques in anesthesia and intensive care.

AI RESEARCH AND CLINICAL MEDICINE

Over the past 25 years, AI research has emerged as a subfield of computer science. The discipline of AI is difficult to define precisely, but can be loosely defined as a scientific field that attempts to develop computer systems that behave in ways that appear to exhibit "intelligence." Implicit in this definition is the assumption that a computer that performs solely numeric computations, such as a statistical analysis, is not performing AI. AI research frequently involves trying to understand how humans reason, often in the presence of ambiguity and incomplete data.

AI researchers have tackled a wide range of very different problems, including playing complex games such as chess, attempting to interpret the meaning of written and spoken English, identifying objects in a visual

From the *Department of Anesthesiology, Stanford University School of Medicine, Stanford, CA, and †Department of Anesthesiology, Yale University School of Medicine, New Haven, CT.

Received Sep 8, 1987, and in revised form Jan 28, 1988. Accepted for publication Feb 5, 1988.

Address correspondence to Dr Rennels, Dept of Anesthesiology, Stanford University School of Medicine, Stanford, CA 94305.

scene, and performing tasks such as medical diagnosis. These are complex tasks, and AI research progress has been a gradual, incremental process. For example, 20 years ago experienced, nonexpert chess players could usually defeat AI chess-playing programs. Today, there are only a small number of players in the world who can defeat the best chess programs [1]. Progress in medical AI research will also be a gradual process.

The field of AI can be somewhat arbitrarily divided into (1) basic AI research, which focuses on theoretical issues, some of which are discussed later in this article; and (2) applied AI research, where the emphasis is on developing EXPERT SYSTEMS to solve real-world problems.

An expert system is a computer system designed to give advice in a domain such as medicine. Such a system is typically constructed by computer specialists who interact with a human expert in the field. These individuals work to make the expert's knowledge concrete and explicit so that it can be incorporated into the computer. The goal is to allow the machine to make expert-level judgments and recommendations in the field or, more practically, to allow the machine to serve as an "intelligent assistant" bringing the expert's encoded knowledge to a user to help the user perform more effectively. A by-product of building medical expert systems is that it forces physicians to analyze and better understand the clinical decision-making process. All the computer systems described in this article are expert systems.

The Nature of Computer Science Research

The AI research projects described in this article are examples of basic computer science research. This type of research is different from the more familiar laboratory and clinical research typically seen in medicine. Computer science research often involves using the computer as a "laboratory" to explore ideas. When confronting a complex computational problem, it is very difficult to develop a well-formulated solution by thinking about the problem in the abstract. The computer provides a vehicle to let the computer scientist obtain feedback concerning his ideas.

As a result, many computer science research projects involve "exploratory programming." In such a project, the computer scientist attempts to gain more understanding about a particular problem, and about how that problem might be solved, by constructing a partial solution in the form of a computer program. The goal of this research is not to develop completely operational, workable computer programs. Rather, the goal is to gain concrete feedback as to what the problems are and how they might be solved.

A successful project, therefore, is often one that gives new insights into the nature of the problem, allows the construction of new tools, or allows one to explore the limitations of existing tools. It is only by an iterative process of exploratory programming that many of the complex problems confronted by AI researchers can be fully understood. Partly for this reason, the systems we describe are not operational systems in clinical use. Rather, they are research prototypes developed to explore the underlying problems of letting the computer assist the physician in patient care.

Medical Computer Science: Confronting the Complexity of Medicine

Medicine has provided a challenging domain for the development of AI-based systems. Indeed, much of the early expert system research during the 1970s was performed in areas of medicine. A wide range of different medical systems were developed and, in the process, a number of AI-based tools were developed [2,3]. During the 1980s, this technology has spread rapidly to a wide range of fields outside of medicine and has generated considerable commercial and industrial activity [4]. At the same time, however, the application of AI in medicine has remained in the research prototype stage.

This situation is somewhat ironic. Medicine has proved to be a productive domain for developing powerful computer-based tools. At the same time, it has been much easier to apply those tools in more structured, less complex domains than medicine, such as in the diagnosis of faults in a manufactured device.

Why is medicine so complex? One way to answer this question is to compare medical diagnosis with diagnosis of faults in, say, an automobile. In a manufactured device, each component has a specific, well-understood function, and the various faults that occur can be understood in terms of these known functions. In medicine, however, the underlying mechanisms of a disease are seldom fully understood. As a result, to perform medical diagnosis, a wide range of diverse knowledge may be required. This knowledge includes (1) known links between diseases and the various signs and symptoms of those diseases, (2) knowledge of possible or partial causal mechanisms of certain diseases, (3) anecdotal case-based knowledge, (4) a wide-ranging knowledge of the clinical literature, and (5) knowledge of social issues relevant to illness and its treatment.

To assist the physician in as sophisticated a fashion as possible, a computer system ideally should be able to integrate all these different types of information. This task is much more difficult than the diagnosis of a manufactured device.

MEDICAL AI RESEARCH: UNDERLYING THEMES

Before a discussion of specific projects, it is useful to understand some of the research themes that medical AI is exploring. This section outlines certain of these themes. Although our emphasis is on AI research in anesthesia and intensive care, we also discuss certain important AI research projects in other areas of medicine.

Knowledge Representation

Perhaps the most fundamental area of basic AI research concerns knowledge representation: developing flexible ways to represent complex real-world knowledge in the machine. A wide range of approaches have been explored. One of these approaches, the use of IF-THEN RULES, merits special discussion, since it has been widely used [5].

An If-Then rule is a simple programming construct consisting of two parts. The first part is an *If* clause, which contains a test. The second part is a *Then* clause, which contains one or more actions the computer is to perform if that test is true.

A well-known early medical expert system that pioneered the use of If-Then rules is MYCIN [6], developed at Stanford to perform diagnosis and recommend treatment for infectious disease. An example of one of MYCIN's If-Then rules follows [6].

If (1) the stain of the organism is grampos,
(2) the morphology of the organism is coccus, and
(3) the growth conformation of the organism is clumps,
Then there is suggestive evidence (0.7) that the identity of the organism is *Staphylococcus*.

This rule is designed to help MYCIN identify a particular organism. The *If* clause tests the variety of features of the organism, and the *Then* clause makes a tentative conclusion as to its identity. One major advantage of the If-Then rules is that they can be translated into English in a fairly straightforward fashion by a rule translation program. (The rule above is shown in its translated form.)

The use of If-Then rules makes it relatively easy for a medical expert to inspect and understand the computer's internal logic. This contrasts with programs written with programming languages such as FORTRAN or BASIC, which often are difficult even for experienced programmers to understand. There are several potential benefits of representing the knowledge in a medical expert system that uses If-Then rules:

1. The medical expert can inspect the knowledge, identify potential errors, and suggest changes.
2. Each rule can be thought of as a small "chunk" of knowledge. In theory, these small chunks of knowledge can be added incrementally to a system, thereby incrementally enhancing its performance.
3. The use of If-Then rules may enhance a system's ability to explain its recommendations, as discussed later.

If-Then rules are used in several of the RULE-BASED SYSTEMS we discuss. In addition to If-Then rules, other widely used AI approaches to knowledge representation include frames [7] and augmented transition networks [8].

Models of Medical Reasoning

The rule-based approach to knowledge representation provides the computer with a set of decision rules, but provides only a fairly superficial understanding of the medical domain. Other AI projects are exploring how a more profound appreciation of the underlying medical domain might be incorporated into the machine to enhance its reasoning.

CAUSAL MODELING. An active area of research involves the development of CAUSAL MODELS to be used by the computer in reasoning about medical problems. A causal model outlines the underlying causal relationships between the components of a complex system, as, for example, between the various components of the cardiovascular system. AI researchers then explore how the model can help the machine reason flexibly about cardiovascular function and pathophysiology.

A pioneering AI system that explored causal models is ABEL [9], which models causal relationships involving acid-base and electrolyte problems and explores how these causal links can help the system reason about the underlying pathophysiology of acid-base and electrolyte disorders. An interesting aspect of ABEL's design is that it demonstrates how a computer can reason causally about a medical problem at different levels of abstraction.

QUALITATIVE CAUSAL MODELING. In addition to being causal models, a number of AI systems are also qualitative models. A QUALITATIVE CAUSAL MODEL is one in which the various system variables are expressed in qualitative terms. For example, the value of a variable (such as blood pressure, cardiac output, or systemic vascular resistance) might be very low, low, normal, high, or very high, rather than a numeric value.

Many AI researchers believe that this qualitative approach allows the program to approximate better the character of expert clinical reasoning, in which medical problems often may be conceptualized in qualitative terms. For example, a physician may believe that "low" cardiac output and "high" blood volume are consistent with a certain type of heart failure. An additional justification for using a qualitative model is that the same (numeric) variable value may be viewed as high in one clinical setting but normal in another. For example, a heart rate of 120 beats/min may be considered normal if blood pressure has recently fallen to abnormally low values, but considered abnormal in other clinical settings.

REASONING FROM THE CLINICAL LITERATURE. Another research project, the Roundsman computer system, explores how computer-based clinical advice might be based not on a pathophysiologic model of the body, but rather on a critical intepretation of the clinical literature [10]. The motivation for this literature-based approach is that drugs and techniques often are used clinically long before underlying mechanisms are fully understood. In such situations, the clinical literature is an important source of knowledge for assessing the risks of alternative management plans. Such assessment requires judgment, since virtually all studies have some methodologic weakness. Also, if clinicians used only studies that perfectly matched their patient, they would rarely find even one such study.

The Roundsman system generates a prose discussion that assesses the difficulties of applying relevant clinical studies (known to the system) to the particular patient being evaluated by the physician. This system has been applied to the management of breast cancer. Anesthesia-related applications might involve, for example, using the clinical literature to help assess the risk of perioperative myocardial infarction in a particular patient with preexisting coronary disease, or the risk of general anesthesia in a hypertensive patient who also has moderately severe chronic obstructive pulmonary disease.

Explanation by the Computer of Its Conclusions

One exciting result of the MYCIN project was that an expert system could, to a certain extent, use its If-Then rules to *explain* its actions and recommendations [11]. For example, if MYCIN has used the rule shown previously to conclude that an organism is probably *Staphylococcus*, and the user asked why, MYCIN could use the English translation of the rule to indicate that the organism was probably *Staphylococcus* because (1) its stain is grampos, (2) its morphology is coccus, and (3) its growth conformation is clumps.

RULE-BASED EXPLANATION. The RULE-BASED EXPLANATION allows the machine to make its internal logic much more transparent to the physician user. Explanations created by translation of If-Then decision rules, however, may not fully address all of the physician's questions. Researchers in the MYCIN project concluded that adequate explanations for the MYCIN system could not be derived from the existing MYCIN program [12]. Computer-based explanation, it was argued, often requires that the computer system have a higher-level, more strategic view of the medical task than is necessary for the computer to achieve good performance.

Another well-known project that addresses the problems of explanation is XPLAIN [13], which was designed to help explain the reasoning of the Digitalis Therapy Advisor system described later. The XPLAIN researchers argue that although the Digitalis Therapy Advisor might be able to perform well by following If-Then rules, robust explanations required that the system have a more profound model of the actions of antidysrhythmic drugs and of heart disease.

CRITIQUING. The development of expert systems that critique patient care represents a different approach to computer-based explanation [14]. Rather than try to tell a physician what to do, a CRITIQUING SYSTEM first asks the physician to outline his proposed approach and then discusses the relative merits of that plan as compared with other approaches that might be reasonable. From the standpoint of computer science research, critiquing can be seen to be a different form of computer-based explanation. By starting with the physician's proposed plan, a critiquing system structures its advice and explanation around the physician's thinking in a natural way. The ATTENDING and VQ-ATTENDING systems, discussed later, illustrate this approach.

Temporal Reasoning

Many expert systems gather information and offer recommendations at a single moment in the patient's clinical course. For example, the INTERNIST-1 system, which has successfully diagnosed difficult Clinicopathological Conference cases from the *New England Journal of Medicine,* diagnoses from one "snapshot" of the patient [15]. Similarly, MYCIN gathers data, diagnoses the cause of an infection, and recommends antibiotics for a given moment in time [6]. Since medical care, especially in the operating room and the intensive care unit, takes place over time, it is desirable to develop computer systems that can evaluate a patient's clinical status and response to treatment over time. Several systems discussed later have addressed this problem.

A related research problem is how to design a computer system to scan a clinical time-ordered database (such as an anesthetic record or intensive care unit flowsheet) and summarize the clinical course, as a physician might do. TEMPORAL MODELING is also a basic research topic in areas of AI beyond medicine [16,17] because, for example, intelligent robots will require this capability if they are to reason about the environment in which they operate.

Integrating Information from Multiple Sources

An active AI research area concerns the problems of assessing information from multiple sources in an integrated fashion. For example, in the intensive care unit, there are (1) real-time physiologic data from a host of monitors, (2) intermittent laboratory data, (3) diverse clinical observations that can be made by inspecting the patient, and (4) a varied patient history that might be relevant to a particular clinical question. Examples are seen in several systems discussed later. The problem of how to design a computer system to integrate this diverse information, taking full advantage of all the information and its interrelationships, is an important area for basic AI research.

Validation and Evaluation

Another major area for basic research concerns how best to "validate" the knowledge in an expert system to assure that it is accurate, complete, and consistent, and how to evaluate the system's clinical efficacy [18]. In addition, a number of projects are exploring how the computer itself can assist actively in the validation process. Researchers have only scratched the surface of the broad spectrum of issues that might be addressed in validating and evaluating medical expert systems.

AI RESEARCH IN ANESTHESIA AND INTENSIVE CARE

We have outlined certain fundamental topics of AI research. The list is far from comprehensive. Next we describe the major systems that explore these research topics in the domains of anesthesiology and intensive care.

To help the reader put the computer systems discussed into perspective, Table 1 presents a capsule outline of each system, indicating the system's domain, together with a rough characterization of certain salient features.

Table 1. Artificial Intelligence Systems in Anesthesia and Intensive Care

System Name	Medical Domain	Selected System Attributes
Harrison's system	Anesthetic management	If-Then rules; explanation capability
ATTENDING	Anesthetic management	Augmented transition networks; critiquing
Ventilator Manager	Ventilatory management	If-Then rules; temporal modeling; qualitative modeling
VQ-ATTENDING	Ventilatory management	If-Then rules; augmented transition networks; critiquing; goal-directed reasoning
Smart respiratory alarms	Ventilatory management	If-Then rules; clinical evaluation
Digitalis Therapy Advisor	Cardiovascular management	Integrating information from multiple sources
Congestive Heart Failure Management System	Cardiovascular management	Truth maintenance system; causal modeling; temporal modeling
Ventricular Arrhythmia Management Advisor	Dysrhythmia management	Temporal modeling; integrating information from multiple sources

AI Research in Anesthesia

HARRISON'S SYSTEM. Harrison and Johnson [19,20] built a system designed to assist an anesthetist in planning intraoperative management. The system produces a set of recommendations based on the clinical details input by the user. These recommendations fall into three categories: (1) avoid action X (e.g., "avoid vasopressors"), (2) use action X (e.g., "use cricoid pressure"), and (3) especially recommend action X (e.g., "peripheral infusion line essential").

In its internal design, Harrison's system uses If-Then rules, most preforming a one-to-one linking of an input value (a clinical feature) to an output value (a recommendation from one of the three categories listed above). There are approximately one thousand such rules. The system's output therefore consists of a set of recommendations, each of which is "triggered" by an input detail. Thus, although this system is rule-based, there is little ability to form complex chains of inferences.

In an initial FORTRAN implementation, there is also the ability to handle contradictory recommendations by writing rules of the form:

If output X and output Y have triggered,
Then replace these by output Z.

Such "interaction rules" were largely unimplemented, however, and in general it is left to the user to interpret the set of recommendations, some of which may be contradictory. The program is viewed as a reminder to the user that certain actions might need consideration. Comprehensive assessment of the reminders is left to the user.

Building on Harrison's initial system, Dodson, Harrison, and Rector [21] developed a "medical treatment planner" for anesthetic management, written in the PROLOG programming language. This system is rule-based with the ability to construct a "chain of inferences." (That is, rule 1 might conclude fact X; fact X might then be used by rule 2 to conclude fact Y, and so on.) The length of such INFERENCE CHAINING is quite short, at most just a few rules. In addition to Harrison's original rules, this PROLOG implementation includes "taxonomic" knowledge and an explanation facility:

1. The taxonomic knowledge allows Dodson and Harrison's system to classify its knowledge, for example, to conclude that a central nervous system depressant premedication could be either a narcotic premedication, an anxiolytic premedication, or a hypnotic premedication. This capability lets the system builders write rules that are more general, for example, a rule mentioning "central nervous system depressant premedication" would apply to all such agents.
2. The explanation facility is in many respects patterned after that of the MYCIN system. The user can ask, for example, why a central nervous system depressant premedication has been suggested, or why it has not been suggested. The system then inspects the various rules that have been triggered and from these rules constructs an explanation. It is also possible for the user to inspect (interactively) all rules that pertain to a particular drug or technique, and also to inspect all conclusions made by the system in the course of a particular consultation.

In summary, Harrison's system is a RULE-BASED CONSULTATION system for anesthetic management, designed primarily to serve as a "reminder" to the anesthesiologist, reimplemented to include taxonomic knowledge and an explanation capability.

THE ATTENDING SYSTEM. The ATTENDING computer system [14,22] explores the critiquing approach to providing computer-based advice for anesthetic management. To use ATTENDING the physician first inputs details describing the patient and the operation, as well as a plan for preanesthetic medication and induction, intubation, and maintenance of anesthesia. This information is entered by a process of menu selection. The computer then assembles an English prose discussion of that plan.

For example, if the physician describes a 34-year-old man with a history of severe asthma who requires an operation in the setting of possible hypovolemia and suggests induction of anesthesia with thiopental, ATTENDING's critique includes the following paragraph. (This paragraph is pieced together from words and sentence fragments tailored to the particular patient and plan.)

> In regard to induction, induction using thiopental could have two possible risks. First, in a patient with asthma, there is the conceivable risk of bronchospasm. Second, there is the possible risk of hypotension in the presence of hypovolemia. An alternative approach to induction using thiopental would be induction with ketamine. This has the advantage of helping avoid hypotension since ketamine is supportive of blood pressure, and of suppressing bronchospasm since ketamine is a bronchodilator.

This paragraph illustrates how ATTENDING is designed to identify and discuss the pros and cons of the physician's proposed plan as well as any reasonable alternatives. It is left to the physician to select the final plan.

ATTENDING represents the alternatives of anesthetic management as a set of networks by using the augmented transition network (ATN) formalism [8]. As shown in Figure 1, an ATN consists of *states* and *arcs* between states. The arcs represent actions, such as general anesthesia, rapid-sequence intubation, or the use of halothane. A *path* through the set of networks represents a complete anesthetic plan for premedication, induction, intubation, and maintenance of anesthesia.

"Higher-level" networks outline anesthetic alternatives such as general anesthesia versus regional anesthe-

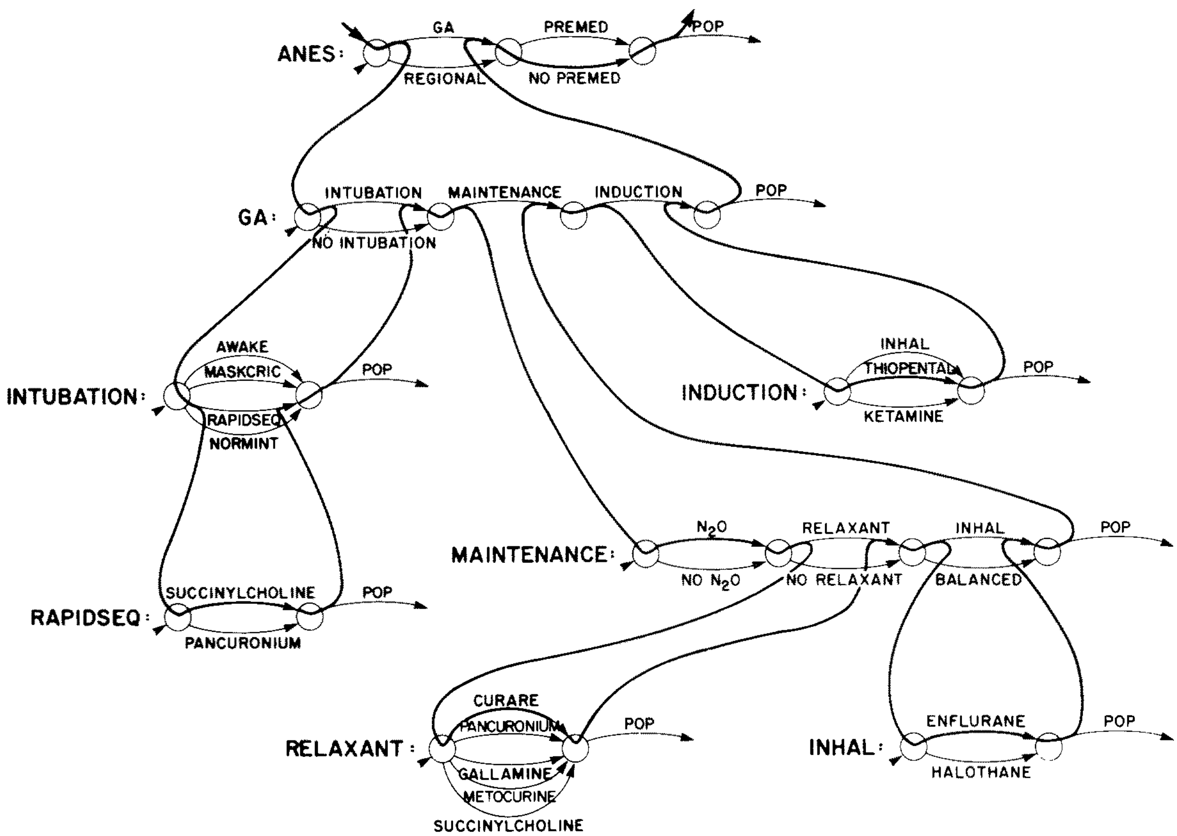

Fig 1. Augmented transition network used by ATTENDING. The bold line *outlines the path through the network of states* (circles) *corresponding to the user's proposed plan. This path indicates a plan to use general anesthesia* (GA), *involving intubation via a rapid-sequence technique* (RAPIDSEQ) *with succinylcholine; maintenance with nitrous oxide* (N_2O), *a relaxant (curare), and the inhalational* (INHAL) *anesthetic enflurane; and induction with thiopental. The exact ordering of the arcs in the network is largely arbitrary. A* "POP" *arc indicates the end of the path through a network.* PREMED = *premedication;* MASKCRIC = *mask induction with cricoid pressure;* NORMINT = *"normal" technique for intubation. (From Miller [22]. Used with permission.)*

sia. "Lower-level" networks then spell out the various subdecisions (and subdecisions within subdecisions) that must be made.

With this ATN scheme, ATTENDING can examine the physician's proposed plan (a particular path through the networks) and analyze alternate paths that might present fewer risks. Each arc has a list of potential risks (of the corresponding technique in different patients). In ATTENDING's analysis, these risks are first assessed roughly. This initial risk analysis may later be refined by "contextual preference rules." The physician's plan, along with any alternatives that ATTENDING concludes worthy of discussion, are then passed to a prose-generation module that assembles an English prose discussion of the plan.

In summary, the emphasis of the ATTENDING system is to center the system's advice around the physician's approach and to bring the salient pros and cons to the physician's attention. It is anticipated that this critiquing approach will prove particularly useful in domains, such as anesthesia, that allow considerable latitude for practice variation and subjective clinical judgment.

AI Research in Ventilator Management

THE VENTILATOR MANAGER SYSTEM. Ventilator Manager (VM) is a prototype system designed to assist physicians and nurses in managing patients receiving mechanical ventilatory support [23,24]. VM is designed to monitor respiratory and related cardiovascular variables in real time. In so doing, it attempts to identify equipment malfunctions and to suggest possible therapeutic interventions. VM was developed between 1978 and 1980, before the current generation of ventilators became widely used.

```
Summary generated at time 15:40

All conclusions:

                                 . . . | . . . . . | . . . . . | . . . . . | . . .
                                    1 2         1 3         1 4         1 5
     BRADYCARDIA[PRESENT]                    = = = = = = = = = =     = =
     HEMODYNAMICS[STABLE]                                                = =
     HYPERVENTILATION[PRESENT]   = = = = = = = = = = = = = = = = = = =   = =
     HYPOTENSION[PRESENT]              = = = = = = =
     Goal Location               C C C C C C C C C C C C C / A A A A A A A A A A A
     Patient Location            V / C C C C C C C C C C C C C C C C C C C C C C C
                                 . . . | . . . . . | . . . . . | . . . . . | . . .
                                    1 2         1 3         1 4         1 5
```

Fig 2. An example of how Ventilator Manager presents a summary of its conclusions about a patient receiving mechanical respiratory support. The computer printout summarizes, over time, the presence (=) or absence of various cardiovascular and respiratory assessments, together with the patient's actual "state" (Patient Location) *and VM's recommended state for the patient* (Goal Location). C = *controlled ventilation;* A = *assisted ventilation;* V = *spontaneous ventilation. (From Buchanan and Shortliffe [eds].* Rule-Based Expert Systems *[18], © 1984, Addison-Wesley Publishing Co, Inc, Reading, MA. Reprinted with permission.)*

VM produces its output in two forms, as shown in Figures 2 and 3. One form is a summary of the conclusions drawn by the system. The other is a sequence of comments and suggestions.

The nuances of this domain were well appreciated by the researchers, and this is reflected in the design issues they chose to pursue:

1. VM represents not just point-estimate values of say, blood pressure, but also temporal trends such as a significant recent rise in the mean arterial pressure.
2. VM assumes that data will be valid for only a certain period of time and will not rely on "old" data to make recommendations.
3. VM uses a qualitative representation for patient variable values such as respiratory rate or expired carbon dioxide. An interesting facet of VM's use of qualitative values is that, for example, a "high" value of expired carbon dioxide might be evaluated differently depending on the particular clinical situation. In general, the system's interpretation of whether particular values are too high, acceptable, etc., is a function of both the patient's clinical state and the physician's therapeutic goals.

```
..  1640..
**  SUGGEST CONSIDER PLACING PATIENT ON T-PIECE IF
**  PA02 > 70 ON FI02 < = .4                           [measure of blood gas status]
**  PATIENT AWAKE AND TRIGGERING VENTILATOR
**  ECG IS STABLE

. .  1650 . . . . 1700 . . . . 1710 . . . . 1720 . . . . 1730 . . . . 1740 . . . . 1750 . .
. .  1800 . .
**  HYPERVENTILATION
**  PATIENT HYPERVENTILATING.
**  SUGGEST REDUCING EFFECTIVE ALVEOLAR VENTILATION.
**  TO REDUCE ALVEOLAR VENTILATION, REDUCE TIDAL VOLUME,
**  REDUCE RESPIRATION RATE, OR
**  INCREASE DISTAL DEAD SPACE TUBING VOLUME

. .  1810 . .
**  SYSTEM ASSUMES PATIENT STARTING T-PIECE

. .  1813 . . . . 1815 . . . . 1817 . .
**  HYPOVENTILATION

. .  1819 . . . . 1822 . .
**  HYPOVENTILATION
```

Fig 3. Sequence of comments and suggestions produced by the Ventilator Manager system regarding a patient receiving mechanical respiratory support. PA02 = *arterial oxygen tension;* FI02 = *inspired oxygen fraction;* ECG = *electrocardiogram. (From Buchanan and Shortliffe [eds].* Rule-Based Expert Systems *[18], © 1984, Addison-Wesley Publishing Co, Inc, Reading, MA. Reprinted with permission.)*

VM uses If-Then rules that were developed in collaboration with intensive care specialists and that fall into four categories:

1. *Status rules.* Status rules make judgments about the patient's cardiovascular and respiratory status, for example, whether the patient's respiratory rate is acceptable. The English translation of an If-Then rule used by VM to check whether hemodynamics are stable is shown below:

Status rule:	Status. Stable hemodynamics.
Definition:	Defines stable hemodynamics based on blood pressures and heart rate.
Applies	to patients on *Volume, CMV, Assist, T-piece.*
Comment:	Look at mean arterial pressure for changes in blood pressure and systolic blood pressure for maximum pressures.
If	(1) heart rate is acceptable, (2) pulse rate does not change by 20 beats/min in 15 minutes, (3) mean arterial pressure is acceptable, and (4) mean arterial pressure does not change by 15 torr in 15 minutes,
Then	the hemodynamics are stable.

2. *Transition rules.* Transition rules attempt to recognize when the patient has been changed to a different ventilator setting or different device. VM cannot assume that physicians will always inform the system when this occurs.
3. *Instrument rules.* Instrument rules attempt to identify artifactual readings, an important concern in the intensive care unit because, for example, ventilator tubing can become reversed or disconnected.
4. *Therapy rules.* Therapy rules recommend action based on the conclusions drawn from the first three categories of rules. In making its recommendations, VM is "deterministic" in that it makes no attempt to weigh the pros and cons of competing options. (The deterministic nature of VM's therapy rules contrasts with VQ-ATTENDING's approach, in which elucidating the pros and cons of various actions is a central research focus.) Types of therapy recommendations include (1) changing ventilator settings, (2) changing ventilator modes, and (3) checking equipment that may be malfunctioning.

Figure 4 outlines VM's approach to therapy recommendation. At any time the system considers the patient to be in one of the therapy states. When VM finds conditions that might allow transition from one state to another, it prints a recommendation that the patient might be started on the therapy corresponding to the destination state. For example, the system can move from the controlled mandatory ventilation (CMV) state in Figure 4 to the assist mode state if certain conditions are met, such as (1) the patient has been on CMV for at least 30 minutes, (2) hypoventilation is absent, and (3) the hemodynamics are stable.

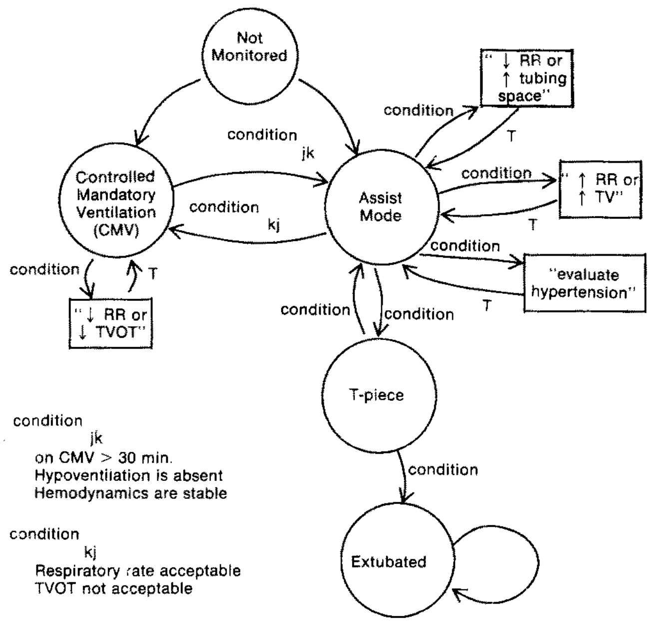

Fig 4. State transition diagram for the Ventilator Manager system. Circles *represent therapy states.* Rules *infer transition (T) between states.* Transient states (boxes) *suggest changes in ventilator setting.* RR = *respiratory rate;* TV = *tidal volume;* TVOT = *tidal volume out.*

Similarly, state transition in the reverse direction, from assist mode to CMV, is possible if other conditions are met, such as (1) respiratory rate is acceptable and (2) the tidal-volume-out (TVOT) value is not acceptable.

Other major states include not monitored, T-piece, and extubated. Finally, transient states are used to suggest changes in ventilator setting, rather than changes in ventilator modes. Transient states can be reached from major states but automatically return the system back to the therapy state. For example, from the assist mode state, certain conditions may move the system to the transient state, "↑ RR [respiratory rate] or ↑ TV [tidal volume]" (see Fig 4). Arriving in this transient state causes the system to print a message: "Suggest increasing respiratory rate or increasing tidal volume." Immediately after this action, the system automatically tra-

Table 2. Suggested Ventilator Changes in Example of VQ-ATTENDING[a]

Variable	Ventilator Setting: Current	Proposed
Inspired oxygen fraction	0.6	0.6
Tidal volume (cm^3)	800	800
Respiratory rate (breaths/min)	8	11
Positive end-expiratory pressure (cm H_2O)	10	12
Ventilator mode	Assist/control	Assist/control
Added dead space (cm^3)	0	0

[a] Adapted from Miller [25].

verses the arc back to the assist mode. Thus, transient states are unlike therapy states in that the system never remains in a transient state, but automatically traverses the return arc.

VM was one of the first medical AI systems designed to help manage a patient over time. In addition, the development of qualitative values for variables was a novel attempt to deal with the fact that numeric values often can be meaningfully interpreted only after assessing the wider clinical context.

THE VQ-ATTENDING SYSTEM. VQ-ATTENDING is a prototype expert system designed to critique aspects of ventilator management [25]. To use VQ-ATTENDING, a physician first enters a limited amount of basic medical information describing a patient who is receiving mechanical respiratory support. The physician also enters current arterial blood gas data, together with current and proposed ventilator settings. VQ-ATTENDING then produces an English prose analysis discussing the appropriateness of the proposed settings.

VQ-ATTENDING extends the critiquing approach beyond anesthesiology to a domain of quite different character. In addition, the VQ-ATTENDING project explores how treatment goals can be made an active part of an expert system's design.

The following example illustrates the goal-directed nature of VQ-ATTENDING's advice. In this example a patient with increased intracranial pressure has arterial blood gas data showing a pH of 7.60, oxygen tension of 40 mm Hg, and carbon dioxide tension of 44 mm Hg. The physician has suggested ventilator changes as shown in Table 2. VQ-ATTENDING's critique of the oxygenation component of this plan is as follows:

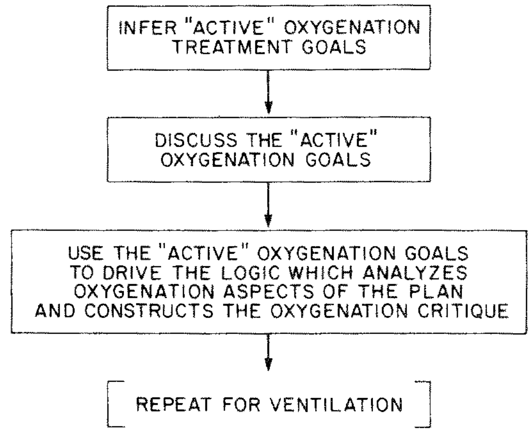

Fig 5. Schematic overview of VQ-ATTENDING's operation. (From Miller [25]. Used with permission.)

In regard to oxygenation, the primary goal in this patient's management is to achieve an adequate PaO_2. Secondary goals are to avoid the risk of oxygen toxicity, and to avoid the risk associated with high PEEP. The urgency of the primary goal, however, may override the secondary considerations.

You have suggested leaving the FiO_2 at 0.6. In view of the urgency of the patient's hypoxemia, it may be worthwhile to use an FiO_2 of 1.0, at least until the patient's oxygenation status has stabilized. Maintaining an FiO_2 greater than 0.6 over extended periods, however, may result in oxygenation toxicity. You have proposed a small increase in PEEP. In a patient with increased intracranial pressure, increased PEEP may exacerbate this problem. In view of this patient's poor oxygenation, an increase in PEEP may ultimately prove necessary, but in this patient, you might first assess the effect of an FiO_2 of 1.0.

As shown in Figure 5, VQ-ATTENDING next produces an analysis of the ventilation component of the plan. Notice that VQ-ATTENDING first discusses the oxygenation treatment goals it considers to apply to the patient's management. It infers these goals with the use of If-Then rules. In the second paragraph, the system then discusses the management choices that might help achieve the goals. The inferred set of treatment goals plays a central role in the internal logic, which produces the critiquing analysis seen in the second paragraph. Thus, treatment goals play a key role both in VQ-ATTENDING's internal analysis and in its prose critique of the physician's plan.

From the standpoint of AI research, the VQ-ATTENDING project makes two main contributions: it extends the critiquing approach to a new domain and it explores a more goal-oriented, strategic design. VQ-ATTENDING separates its knowledge of ventilator management into two parts: (1) strategic knowledge about treatment goals (e.g., to achieve adequate arterial oxygenation) and (2) tactical knowledge about the management alternatives used to achieve those goals (e.g., to increase FiO_2).

VQ-ATTENDING's goal-directed design raises several questions: First, are there particular medical domains where strategic reasoning of this sort is especially useful? Second, how should a computer system deal with possible conflicts between goals? Third, what should the system do if the physician has different goals from those the system considers appropriate? (One approach would be to let the system critique the physician's plan at the level of goals as well as at the level of tactics.) The prototype implementation of VQ-ATTENDING is a first step in exploring these issues. It seems clear, however, that the ability for a computer to counsel a physician regarding overall strategy as well as tactics is highly desirable in many areas of medicine.

SMART RESPIRATORY ALARMS. Scientists at Pacific Medical Center in San Francisco have developed a *smart respiratory alarm* system [26]. This system's knowledge, expressed by using If-Then rules, monitors the signals generated by a bedside ventilator. The program is designed to recognize twenty-three separate alarms, each belonging to one of three categories:

1. *Monitoring equipment malfunctions.* For example, one rule (presumably reflecting the characteristics of the ventilator used) states:

   If (1) the gas sample tube is connected,
   (2) FiO_2 and expiratory oxygen fraction (FeO_2) are not above 0.21, and
   (3) oxygen uptake = 0,
   Then the system should print the message "No O_2 signal."

2. *Ventilator-related alarms.* For example, a rule designed to detect endotracheal cuff leaks states:

   If (1) the patient is being mechanically ventilated,
   (2) the expiratory tidal volume is less than three-fourths of the inspiratory tidal volume, and
   (3) the expiratory tidal volume decreased over 10% in the time interval since the last measurement,
   Then the computer should print the message: "Patient not getting full volume. Check for cuff leak."

3. *Patient-related alarms.* For example, patients recovering from the hypothermia induced during heart surgery may be shivering or frightened. One rule looks for these possibilities by scanning for extreme increases in metabolic rate. The rule states:

   If the peak oxygen uptake and the peak carbon dioxide production each exceed 500 ml/min,
   Then the computer should print the message: "Oxygen uptake is high. Is patient frightened or shivering?"

Ventilator signals are obtained from a gas sampling system placed between the Y of the ventilator tubing and the patient. Primary data gathered include flow, pressure, and oxygen and carbon dioxide tensions at the outlet of the endotracheal tube. These primary data elements are processed to determine secondary monitoring variables, such as tidal volume inspired, tidal volume expired, respiratory rate, PEEP, and also an index of how much the patient is fighting the ventilator.

Performance of the smart alarm system was evaluated in 157 postcardiac surgery patients over a six-month period. The nurses caring for those patients were interviewed to determine their opinions of the 476 alarms that had occurred. The types of alarms those nurses considered most useful occurred infrequently (fewer than six times per month). The smart alarm was viewed as a useful backup in those infrequent situations.

AI Research in Cardiovascular Management

THE DIGITALIS THERAPY ADVISOR. The Digitalis Therapy Advisor [27] is designed to assist in the clinical use of digitalis. The program differs from previous programs that have dealt with digitalis therapy [28,29] because the Digitalis Therapy Advisor focuses on clinical endpoints rather than on serum levels of digitalis. The desired therapy is defined by clinical goals: improvements in the clinical problem (therapeutic benefits) balanced against toxic manifestations.

The program is built around a *patient-specific model* (PSM), which includes such variables as the patient's age, weight, and the clinical reason for using digitalis. The program uses a pharmacokinetic equation to predict the body stores of digitalis at a given time. (If clinical information indicates that the digitalis level is much lower than predicted by the equation, the program attempts to correct the PSM by, for example, altering the oral absorption variable.)

Subsequent dosing adjustments are made primarily on the basis of clinical feedback rather than on the basis of measured serum levels. If the clinical indication for administering digitalis is heart failure, the system will ask about signs indicating clinical improvement or worsening (e.g., pulmonary congestion). On the other hand, if the initial reason for giving digitalis is a dysrhythmia, an entirely different set of clinical variables is the focus for assessing response to therapy (e.g., slowing of atrial fibrillation). AI techniques are especially relevant in this problem because these clinical findings are not strictly numeric, and their evaluation is a matter of expert judgment rather than, say, pharmacokinetics.

The Digitalis Therapy Advisor was evaluated in 1976 by collecting patient histories and progress notes for one month and running the program on these patients retrospectively. Nineteen patients were reviewed, representing every patient receiving digitalis in the cardiology service of Tufts-New England Medical Center Hospital (excluding patients on routine, stable maintenance schedules). The computer program detected all 4 patients in whom toxicity developed and always detected it before the physicians handling the patients had. There were no false positives (that is, the program's suggesting toxicity when there was none) among the other 15 patients. These encouraging results must not obscure the fact that the program's focus is narrow (as is the case with many experimental AI programs). For example, the program does not take into account many coexisting diseases or conditions (e.g., pacemakers) that may complicate the clinical situation.

The principal research contribution of the Digitalis Therapy Advisor then, is not simply that it follows the patients over time, but that it uses clinical information as input. This requires that the program evaluate clinical findings as well as straight numeric measurements and that the pursuit of therapeutic benefit is tempered by any clinical sign(s) of toxic manifestations. This system was one of the first to smoothly integrate "soft" clinical information with a mathematical model of pharmacokinetics.

CONGESTIVE HEART FAILURE MANAGEMENT. A developmental system at the Massachusetts Institute of Technology uses a causal, qualitative model of human pathophysiology to provide advice for the management of congestive heart failure (CHF) [30–33]. To use the program, the physician enters clinical findings and laboratory data. (It is necessary to use the restricted vocabulary of the computer program rather than free-form text, but the information can be entered in whatever order the user desires.) This information is used to set qualitative values for variables in the physiologic model (discussed in more detail below) and to trigger If-Then rules that may conclude that certain disease states are present. The system then uses the physiologic model to assess the completeness and consistency of possible diagnoses and to search for interventions that might improve cardiac function. Figure 6 gives an overview of the system's operation.

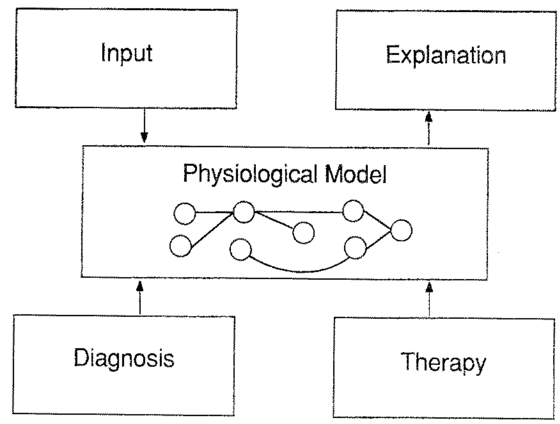

Fig 6. Outline of the design of the congestive heart failure system. A qualitative physiologic model forms the backbone of the system. The model receives input describing a particular patient, and is then able to assist (with appropriate explanations) in assessing diagnosis and therapy for that patient. (Adapted from Long et al [33] © 1984 IEEE.)

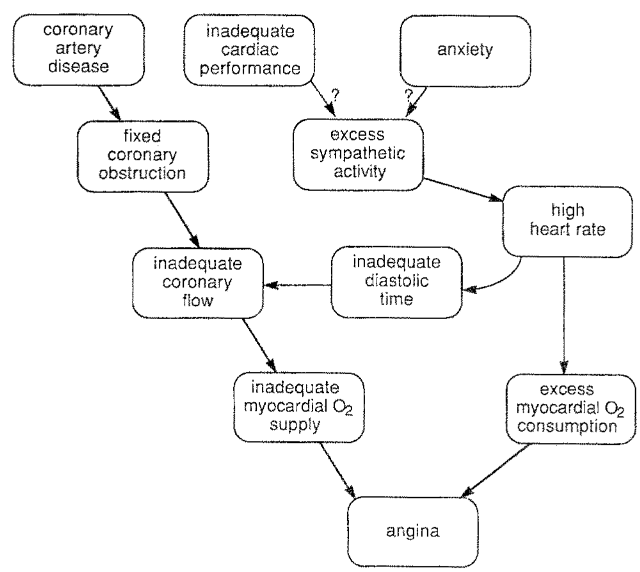

Fig 7. Small subset of causal nodes and links used by the congestive heart failure management system. Connecting arrows represent the influence of one pathophysiologic state (node) on another. In this example, the presence of angina is known at the outset and the system tries to confirm or deny (causally) contributing problems. (Adapted from Long et al [33] © 1984 IEEE.)

The CHF management system's causal model is represented by nodes (physiologic states of the cardiovascular system relevant to heart failure) that are linked together by relations (Fig 7). There are causal relations between states and time dependency relations between states. There are several hundred nodes in the system, only a subset of which are "active" (relevant to the patient described) at any one time. The CHF management system searches along causal chains (between abnormal findings and possible causes) to find a coherent interpretation of the patient's pathophysiology.

Management advice is also generated by inspection of these causal chains for possible therapeutic interventions. In so doing, the computer can vary a variable (e.g., increase peripheral vascular resistance) and automatically propagate the effects of this change throughout the model. In this way, the computer itself can experimentally evaluate the effects of different interventions in a particular patient. As effects are propagated, the justification for any variable change is stored by a module called the *truth maintenance system*. This allows the system to explain its reasoning. An additional role of the truth maintenance system is to identify ambiguities and contradictions in the model's variable values. The CHF management system will note these ambiguities and try to resolve them.

Several features of the system are worthy of special note:

1. *The physiologic model is a qualitative model.* All variable values are qualitative rather than numeric. For example, the value of cardiac output may be "low" or "high" rather than a number.
2. *Temporal relationships are central to the program's reasoning.* Temporal relationships are integrated into the causal model. In addition, just as cardiovascular variables are represented qualitatively, temporal duration also is represented by qualitative values. Qualitative temporal values are useful because, for example, although the exact time when low cardiac output started may not be known, it may be possible to say that it has been present for days, but less than one week. Even such inexact time information is often useful in diagnostic and therapeutic reasoning. In addition to duration, to determine causality it is important to represent the temporal order of events, and for this reason the system superimposes a partial ordering on temporal events, even when exact times are unknown.
3. *Cost and risk.* The CHF management system also allows cost and risk to play an active role in resolving diagnostic ambiguities and in recommending therapy. For example, the diagnostic module assesses the costs and risks of different measurements to help select those that might clarify the diagnosis at minimum risk. (Thus, less invasive procedures are preferred). These utility considerations are rudimentary, but they demonstrate that the system designers recognize the importance of cost and risk considerations in medical reasoning.

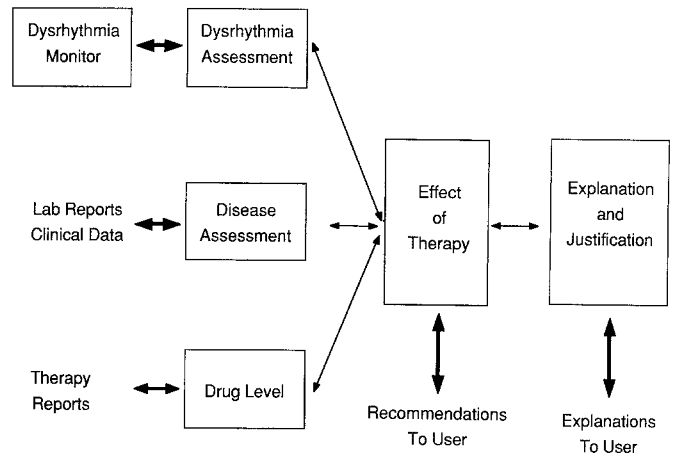

Fig 8. The Ventricular Arrhythmia Management Advisor system. (Adapted from Long et al [35] © 1983 IEEE.)

The CHF management system is currently under active development. From an AI research standpoint, this project is exploring the use of a qualitative causal model to drive a system's reasoning about management of heart failure and the use of temporal information about clinical findings to augment the causal reasoning.

VENTRICULAR DYSRHYTHMIA MANAGEMENT. The Digitalis Therapy Advisor is the precursor to the Ventricular Arrhythmia Management Advisor (VAMA) project [34,35]. As illustrated in Figure 8, VAMA is designed to accept real-time electrocardiographic input, as well as a large variety of other clinical information, and offer recommendations for the management of ventricular dysrhythmias.

Like the Digitalis Therapy Advisor, VAMA is designed around the central concept of using clinical information as feedback to guide drug therapy. The tradeoff between therapeutic benefit and toxicity plays a major role in this program's analysis as well. VAMA generalizes the approach of the Digitalis Therapy Advisor in at least three ways:

1. VAMA considers a wider range of cardiac drugs than just digitalis.
2. By using a pharmacokinetic model, VAMA examines not only steady-state drug levels (as does the Digitalis Therapy Advisor) but also temporal pharmacokinetic trends. For example, VAMA might consider the time elapsed since a lidocaine bolus and conclude that the plasma concentration is in a slowly

increasing phase and has not yet reached the steady-state phase. This allows VAMA to make dosing recommendations at times preceding the steady-state phase.

3. VAMA considers diverse sources of information in its determination of therapy recommendations: (1) cardiac rhythm analysis (one component of which is a commercial dysrhythmia monitor), (2) the patient's clinical history, (3) clinical laboratory data (e.g., serum drug levels), and (4) a clinician's observations of the patient's state.

The remainder of this discussion focuses on two important aspects of the VAMA system: its integration of multiple sources of knowledge and its ability to consider temporal trends.

The integration of information from multiple sources is a major challenge for AI research. Anesthesiologists and intensive care physicians are acutely aware that multiple sources of data must be continually integrated into an overall patient assessment. The anesthesiologist continually "scans" a variety of data sources: the patient, electrocardiogram, blood pressure, ventilator settings, oximeter values, actions of the surgeon, condition of the operating field, volatile agents, time elapsed since intravenous drug administration, patient history, and so forth. The experienced clinician learns to integrate these, and also to prioritize abnormal observations. AI research such as the VAMA project seeks to understand this human ability in order to design computer systems that can assist the physician in this process rather than simply generating yet more data to be analyzed.

In VAMA, the diverse sources of information are integrated by intermediate subprograms. For example, the patient history, laboratory reports, and user-supplied information about the current clinical state are integrated into a PSM. A therapy advisor module then draws upon the PSM, the system's pharmacokinetic model, and the results of the dysrhythmia evaluation unit to make recommendations.

VAMA's ability to consider the temporal trends of plasma drug levels allows it to handle intravenous infusion rates as well as drug boluses. For example, in one actual patient the cardiac care unit team observed that dysrhythmias persisted while a lidocaine drip was running, and they responded to this situation by increasing the drip rate. When VAMA considered this patient retrospectively, it used temporal aspects of the pharmacokinetic model to recognize correctly that the plasma concentration was probably still rising slowly toward the steady-state level. To reach the steady-state concentration sooner, it was best to bolus rather than to increase the infusion rate. (After the cardiac care unit team had increased the drip rate, the serum level eventually surpassed the therapeutic range and the patient became somewhat toxic.)

In summary, VAMA is a developmental AI research project with the goal of building a computer system for dysrhythmia management of cardiac care unit patients. VAMA generalizes the capabilities of the Digitalis Therapy Advisor in several ways. The salient research issues are the ability to consider temporal trends and the improved capability to integrate diverse sources of information into a comprehensive clinical picture.

CURRENT STATUS OF THE FIELD

It is important to emphasize that the computer systems described here are research prototype systems. None are in clinical operation. Some of the systems have been tested on retrospective data from a small number of patients, or by limited testing in the clinic.

Although the development of practical systems is certainly an ultimate goal, the immediate goal of these projects has been to explore the AI research problems involved in giving computer-based clinical advice and to begin to develop solutions. Major problems remain at both the research and practical level, many of which have barely been touched. These problems include (1) linking the systems to real-world data collection where many practical issues, such as dealing with artifacts [36], are extremely important; (2) dealing comprehensively with the need for sophisticated, "intelligent" alarms [37]; (3) tailoring the systems to the demands of the clinical environment, so the systems are human-engineered to maximize their efficacy; (4) dealing fully with the variability of medicine, so the systems will deal appropriately with the full range of coexisting disease, concurrent treatment, etc., that will be encountered; (5) testing and validating decision rules and algorithms; and (6) accommodating practice variations among different experts and institutions.

Two recent articles provide thoughtful discussions of certain of these issues. Rampil [36] gives an overview of computer-based detection of artifact, surveying work done in monitoring different real-time patient data streams, discussing the general problems which must be dealt with, and suggesting that AI-oriented techniques might help in the process. Beneken and Gravenstein [37] propose a model for conceptualizing the design of sophisticated alarms using systems engineering concepts. Both of these articles are good illustrations of the type of thinking required to bridge the gap between the practical problems posed by real-world patient monitoring and the more theoretical issues currently being dealt with by basic AI research.

Researchers in the field of medical AI believe that basic research is required before many of these practical issues can be confronted in a satisfactory way. We have described several ongoing basic research projects that are confronting these fundamental issues in the anesthesia and intensive care settings.

SUMMARY

Medicine is a complex and demanding domain for the implementation of computer-based advisors. There is a broad range of different types of information that can be brought to bear on medical problems, including fundamental biomedical principles, clinical observations linking disease states to clinical findings, anecdotal case-based knowledge, and critical interpretation of the clinical literature. Granting the complexity and diversity of medical knowledge, it should not be surprising that the development of robust, sophisticated computer-based advisors poses challenging problems. We have outlined a set of research projects that are beginning to confront these problems in the areas of anesthesiology and intensive care.

APPENDIX: AI PROGRAMMING LANGUAGES AND HARDWARE

A discussion of AI is incomplete without mentioning the LISP (*List Processing*) programming language, widely used by AI researchers [38]. LISP was one of the first languages designed for symbolic (nonnumeric) programming. A second computer language that has been used more recently for AI programming is PROLOG [39].

While LISP has been the central programming tool for many AI researchers for the past 20 years, the hardware used has changed. In the 1970s, most AI work was performed with large mainframe computers, time-shared among several users. In the 1980s, there has been a migration of these projects to scientific workstations. These workstations are powerful, single-user computers, which have recently cost $60,000 or more. This price is dropping dramatically, and scientific workstations will soon become considerably more affordable.

Dr Rennels' work is supported in part by National Institutes of Health grant LM07033 from the National Library of Medicine. Dr Miller's work is supported in part by National Institutes of Health grants R01 LM04336 and T15 LM07056 and National Institutes of Health contract N01 LM63524 from the National Library of Medicine.

GLOSSARY

ARTIFICIAL INTELLIGENCE (AI) A subfield of computer science that can be loosely defined as the discipline of developing computer systems that exhibit "intelligent" behavior.

CAUSAL MODELING An area of AI research that is exploring how underlying causal relationships in a physical or biologic system can be represented so that the computer itself can inspect those causal relationships and reason about their various implications.

CRITIQUING SYSTEM An expert system that critiques its user's approach to a particular problem rather than didactically attempting to tell the user what to do.

EXPERT SYSTEM A computer consultation system designed to operate in a real-world domain such as medicine, embodying the expertise of human specialists in that field.

IF-THEN RULE A conceptually simple yet powerful knowledge representation technique often used in AI programming. An If-Then rule consists of two parts. The *If* clause is a test. The *Then* clause is an action the computer will perform if that test is true. Often the action involves making a conclusion (an "inference") that may later be tested by the *If* clause of other If-Then rules.

INFERENCE CHAINING If the conclusion made by the *Then* clause of one rule is tested by the *If* clause of another rule, then those two rules can chain together sequentially to form an inference chain.

QUALITATIVE CAUSAL MODELING A type of causal modeling that is being used to explore how the various state variables in a causal model might be expressed in qualitative terms (e.g., "normal," "low," and "high"), rather than in numeric values.

RULE-BASED EXPLANATION This occurs when an expert system explains the questions it asks its user or the recommendations it makes, by using, for example, its If-Then rules as the basis of the explanation.

RULE-BASED CONSULTATION SYSTEM An expert computer system constructed by using If-Then rules to represent the system's knowledge about its domain.

TEMPORAL MODELING An area of AI research that is exploring how best to represent temporal events (events occurring over time) and relationships between those events so that the computer can reason about those events.

REFERENCES

1. Leithauser B. The space one breath. The New Yorker 1987, Mar 9:41–73
2. Clancey WJ, Shortliffe EH, eds. Readings in medical artificial intelligence. Reading, MA: Addison-Wesley, 1984
3. Szolovits P, ed. Artificial intelligence in medicine. Boulder, CO: Westview Press, 1983

4. Buchanan BG. Expert systems: working systems and the research literature. Expert Systems 1986;3:32–51
5. Buchanan BG, Shortliffe EH, eds. Rule-based expert systems. Reading, MA: Addison-Wesley, 1984
6. Shortliffe EH. Computer-based medical consultations: MYCIN. New York: American Elsevier, 1976
7. Minsky M. A framework for representing knowledge. In: Winston P, ed. The psychology of computer vision. New York: McGraw-Hill, 1975:211–277
8. Woods WA. Transition network grammars for natural language analysis. Commun Assoc Comput Machinery 1970;13:591–606
9. Patil RS, Szolovits P, Schwartz WB. Causal understanding of patient illness in medical diagnosis. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence. Los Altos, CA: William Kaufmann; 1981: 893–899
10. Rennels GD. A computational model of reasoning from the clinical literature. In: Proceedings of Symposium for Computer Applications in Medical Care. New York: IEEE; 1986:373–380
11. Scott AC, Clancey WJ, Davis R, Shortliffe EH. Methods for generating explanations. In: Buchanan BG, Shortliffe EH, eds. Rule-based expert systems. Reading, MA: Addison-Wesley, 1984:338–362
12. Clancey WJ, Letsinger R. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In: Clancey WJ, Shortliffe EH, eds. Readings in medical artificial intelligence: the first decade. Reading, MA: Addison-Wesley, 1984:361–381
13. Swartout WR. Explaining and justifying expert consulting programs. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence. Los Altos, CA: William Kaufmann; 1981:815–822
14. Miller PL. Expert critiquing systems: practice-based medical consultation by computer. New York: Springer-Verlag, 1986
15. Miller RA, Pople HE, Myers JD. INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine. N Engl J Med 1982;307:468–476
16. Allen JF. Maintaining knowledge about temporal intervals. Commun Assoc Comput Machinery 1983;26:832–843
17. Allen JF. Towards a general theory of action and time. Artificial Intelligence 1984;23:123–154
18. Buchanan BG, Shortliffe EH. The problem of evaluation. In: Buchanan BG, Shortliffe EH, eds. Rule-based expert systems. Reading, MA: Addison-Wesley, 1984:571–588
19. Harrison MJ, Johnson F. Computer-assisted decision making in anaesthesia. Br J Anaesth 1980;52:629
20. Harrison MJ. Codifications of anesthetic information for computer processing. J Biomed Eng 1981;3:196–199
21. Dodson DC, Harrison MJ, Rector AL. A prototype knowledge-based medical treatment planner. In: Proceedings BCS Expert Systems, 1983
22. Miller PL. Critiquing anesthetic management: the ATTENDING computer system. Anesthesiology 1983; 58:362–369
23. Fagan LM, Kunz JC, Feigenbaum EA, Osborn JJ. Extensions to the rule-based formalism for a monitoring task. In: Buchanan BG, Shortliffe EH, eds. Rule-based expert systems. Reading, MA: Addision-Wesley, 1984:397–423
24. Fagan LM, Shortliffe EH, Buchanan BG. Computer-based medical decision making: from MYCIN to VM. In: Clancey WJ, Shortliffe EH, eds. Readings in medical artificial intelligence. Reading, MA: Addison-Wesley, 1984: 241–255
25. Miller PL. Goal-directed critiquing by computer: ventilator management. Comput Biomed Res 1985;18:422–438
26. Mitchell RR, Feihl F, Osborn JJ. A clinical evaluation of a knowledge-based respiratory alarm system. In: Proceedings of the Second Annual Symposium on Computing in Anesthesia and Intensive Care, Rotterdam, 1983:45
27. Gorry GA, Silverman H, Pauker SG. Capturing clinical expertise: a computer program that considers clinical responses to digitalis. Am J Med 1978;64:452–460
28. Jelliffe RW, Buell J, Kalaba R. Reduction of digitalis toxicity by computer-assisted glycoside dosage regimens. Ann Intern Med 1972;77:891–906
29. Sheiner LB, Rosenberg B, Melmon K. Modelling of individual pharmacokinetics for computer-aided drug dosage. Comput Biomed Res 1972;5:441–459
30. Long WJ, Naimi S, Criscitiello MG. A knowledge representation for reasoning about the management of heart failure. In: Proceedings of the IEEE Conference on Computers in Cardiology. New York: IEEE; 1982:373–376
31. Long WJ, Russ TA. A control structure for time dependent reasoning. In: Proceedings of the International Joint Conference on Artificial Intelligence. Los Altos, CA: William Kaufmann; 1983:230–232
32. Long WJ. Reasoning about state from causation and time in a medical domain. In: Proceedings of the National Conference on Artificial Intelligence. Los Altos, CA: William Kaufmann; 1983:251–254
33. Long WJ, Naimi S, Criscitiello MG, et al. An aid to physiological reasoning in the management of cardiovascular disease. In: Proceedings of the IEEE Conference on Computers in Cardiology. New York: IEEE; 1984:3–6
34. Russ TA. A knowledge-based approach to ventricular arrhythmia management. In: Proceedings of the International Conference on Cybernetics and Society, 1982:10–14
35. Long WJ, Russ TA, Locke WB. Reasoning from multiple information sources in arrhythmia management. In: Proceedings of the IEEE-83: Frontiers of Engineering and Computing in Health Care. New York: IEEE; 1983:640–643
36. Rampil IJ. Intelligent detection of artifact. In: Gravenstein JS, Newbower RS, Ream AK, Smith NT, eds. The automated anesthesia record and alarm systems. Boston: Butterworths, 1987:175–190
37. Beneken JEW, Gravenstein JS. Sophisticated alarms in patient monitoring: a methodology based on systems engineering concepts. In: Gravenstein JS, Newbower RS, Ream AK, Smith NT, eds. The automated anesthesia record and alarm systems. Boston: Butterworths, 1987:211–228
38. Winston PH, Horn BK. Lisp. Reading, MA: Addison-Wesley, 1984
39. Clocksin WF, Mellish CS. Programming in PROLOG. New York: Springer-Verlag, 1981