CRITIQUING BASED ON COMPUTER-STORED MEDICAL RECORDS

(BEKRITISEREN VAN HET MEDISCH HANDELEN OP GROND VAN
GEGEVENS IN ELEKTRONISCHE MEDISCHE DOSSIERS)

Proefschrift

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de rector magnificus
Professor Dr. C.J. Rijnvos
en volgens besluit van het College van Dekanen.

De openbare verdediging zal plaatsvinden op
woensdag 6 maart 1991 om 15.45 uur.

door

JOHAN VAN DER LEI
geboren te Sneek

Promotiecommissie

Promotores	:	Prof.Dr.Ir. J.H. van Bemmel
		Prof.Dr. E. van der Does
Referent	:	Dr. R.F. Westerman
Overige leden	:	Prof.Dr. A. Hofman
		Prof.Dr. A.J. Man in't Veld
		Prof.Dr. A. Hasman

Acknowledgment

The study was supported by a grant from the Netherlands Heart Foundation.

Financial support by the Netherlands Heart Foundation for the publication of this thesis is also gratefully acknowledged.

Ad majorem Dei gloriam

Voor Auli, Tellervo, Gerben,
en Anna-Marie.

VOORWOORD

In dit proefschrift staat centraal de vraag hoe computers, op basis van gegevens uit geautomatiseerde medische dossiers, kritiek kunnen leveren op het handelen van de arts met als doel het voorkómen van mogelijke tekortkomingen in die behandeling. Computersystemen zijn gebaseerd op modellen; het ontwikkelen van een systeem voor een bepaalde toepassing betekent dan ook het ontwikkelen van een model voor die toepassing. Het ontwikkelen van een kritiek-systeem behelst dan ook het ontwikkelen van een model dat het leveren van kritiek beschrijft. In dit proefschrift wordt gepoogd een aanzet te geven tot het ontwikkelen van een dergelijk model. In deze studie hebben we ons beperkt tot het leveren van kritiek op de behandeling van essentiële hypertensie in de eerstelijns gezondheidszorg. Ook de implementatie (het operationeel maken op een computer) van het model en de evaluatie van het resulterende systeem beperken zich tot de behandeling van essentiële hypertensie. Het is onze verwachting dat het in dit proefschrift beschreven model niet slechts geschikt is voor het leveren van kritiek op de behandeling van essentiële hypertensie, maar dat het model breder toepasbaar is. Verder onderzoek zal moeten leren in hoeverre het door ons beschreven model aangepast zal moeten worden wanneer het gebruikt wordt bij het bekritiseren van andere medische behandelingen.

Het uitvoeren van deze studie is enerzijds een eenzame periode geweest, maar anderzijds een periode waarin vele mensen mij geholpen en gesteund hebben. Eenzaam omdat het uiteindelijk de worsteling is met je eigen beperkingen, en steun omdat velen de worsteling mogelijk hebben gemaakt. Het is dan ook een goed gebruik dat een promovendus in een voorwoord de mogelijk illusie wegneemt dat de onderhavige studie uitsluitend door de promovendus zelf mogelijk is gemaakt en uitgevoerd.

Allereerst wil ik die huisartsen bedanken die de moed hadden om hun handelen door andere artsen te laten beoordelen. Deze huisartsen hebben het onderzoek mogelijk gemaakt door zich kwetsbaar op te stellen. Ze hebben laten zien dat ze behoren tot de categorie artsen die bereid zijn na te denken over hun eigen handelen, ook als dat impliceert dat hun eigen handelen ter discussie word gesteld. Deze bereidheid om zichzelf aan een dergelijke diepgaande intercollegiale toetsing te onderwerpen verdient respect en navolging.

Ook de huisartsen en specialisten die bereid waren de medische dossiers van hun collega's van kritiek te voorzien wil ik bedanken. De decimeters papier in mijn ladenkast zijn de stille getuigen van de vele uren vrije tijd door hen aan dit onderzoek besteed.

Mijn dank ook aan Frans Westerman die zich vanaf het eerste begin heeft ingezet voor het welslagen van deze studie. Ik heb in de afgelopen jaren nooit tevergeefs een beroep op hem gedaan.

De Nederlandse Hartstichting heeft het onderzoek financieel ondersteunt, en maakte daarmee de komst van Mees Mosseveld en Rosa Scholte mogelijk. De hulp en ondersteuning die Rosa vanuit het secretariaat heeft geboden en de kundigheid waarmee Mees de voor dit onderzoek benodigde programmatuur verder heeft ontwikkeld worden door mij zeer gewaardeerd. Speciale dank ben ik dan ook verschuldigd aan al de vrijwilligers die het de Nederlandse Hartstichting mogelijk maken wetenschappelijk onderzoek te ondersteunen.

Ik ben me er van bewust dat ik de schouders van collega's binnen de vakgroep Medische Informatica heb gebruikt om op te staan; zij hebben de omgeving gecreëerd waarin dit onderzoek kon worden uitgevoerd. Door hun positieve kritiek en prettige samenwerking hebben zij me een omgeving geboden waarin ik me altijd thuis heb gevoeld. Met name wil ik Jan van Bemmel bedanken voor de positieve ondersteuning gedurende de afgelopen jaren.

Een speciaal woord van dank voor Mark Musen. Beste Mark, bijna elke pagina van dit proefschrift is door jou van kritische opmerkingen voorzien. Niet alleen voor de inhoud van onze vele discussies, maar vooral voor de wijze waarop jij je vriendschap in woord en daad hebt getoond ben ik je zeer erkentelijk.

Mijn ouders wil ik danken voor de loyale hulp en steun die ik altijd heb mogen ontvangen, en voor de wijze waarop jullie mij altijd hebben gestimuleerd.

Lieve Auli, Tellervo, Gerben en Anna-Marie. Jullie zijn enerzijds een bron van inspiratie, en anderzijds helpen jullie mij om mijn werk in het juiste perspectief te plaatsen. Het is aan jullie dat ik dit proefschrift op wil dragen.

CONTENTS

CHAPTER 1

Introduction

1.1 INTRODUCTION

Many medical decision-support systems rely on a consultation model for their interaction with the user. In the consultation model the program serves as an advisor, accepting patient-specific data, asking questions, and generating advice for the user about diagnosis or therapeutic management [1-3]. Certain workers in medical informatics have argued that, for some medical domains, critiquing the decisions of a physician is a preferred approach in providing decision support. In this critiquing model the physician submits to the program, in addition to patient-specific data, the decisions he intends to make. The program evaluates these decisions and expresses agreement or suggests alternatives [4]. Other researchers have stressed the importance of integrating consultation systems with routine data-management functions within a medical office or institution. When decision-support systems are integrated with data-management systems, provision of decision support can be viewed as a byproduct of the data-management activities [5-7]. Other workers have attempted to combine the critiquing approach with data-management systems, resulting in systems that, from the viewpoint of the physician, act as automated medical records, but that "behind the scenes" evaluate the decisions of the physician and, if necessary, suggest reasoned alternatives [8,9].

This study describes the design, implementation, and evaluation of a critiquing system that relies on automated medical records for its data input. The purpose of the system is to offer comments to general practitioners on the treatment of hypertension. The system, HyperCritic, has access to the data stored in a primary-care information system that supports a computer-stored medical record. A major restriction that we have imposed on HyperCritic is that the program must rely solely on this automated medical record for data input; a consultation-style interaction with the user is avoided.

1.2 GENERATION OF COMPUTATIONAL MODELS

Underlying any computational system is a model of the task domain of that system and of the methods by which problems in that domain are addressed. For example, a hierarchical database presumes a data model based on hierarchical relationships among data elements, whereas a relational database permits system builders to describe a variety of tabular relationships among data elements. The form of the model subsequently dictates the terms and

relationships used in the system. In the case of a relational database, the relational model is expressed in terms of tables in the database and of the operations (for example, select, project, and join) that manipulate the data stored in these tables. In contrast to the explicit data models that undergird database systems, the domain models underlying decision-support systems often are implicit. The behavior of such systems is described in terms of the symbols (for example, rules, frames, or objects) and the inference strategies that manipulate those symbols (for example, forward chaining, backward chaining, or belief update in causal probabilistic networks) [10].

An important perspective articulated by Newell [11] is that knowledge is an abstraction that can be separated from the symbols that are used to represent the knowledge. Knowledge, in the view of Newell, is a set of goals and the behavior potentially needed to achieve those goals. Knowledge itself can never be written down; it can only be observed as an activity. This distinction between knowledge (at what Newell refers to as the *knowledge level*) and the symbols used to represent knowledge (the *symbol level*) allows us to distinguish our goals for an intelligent system from the language that we use at the symbol level to represent these goals. Thus, knowledge-level analysis of an application task specifies the behaviors that are required to solve a problem in the world; analysis of a knowledge base at the symbol level specifies the computational mechanisms needed to model the requisite behavior. Researchers in artificial intelligence (AI) increasingly agree that it is important to understand a domain task in terms of its knowledge-level specifications before proceeding to a symbol-level implementation. Although there is little consensus about how to go about describing domain tasks at the knowledge level, the goal becomes to understand a system's behavior in terms of an abstract model, rather than by means of a specific set of notations [11,12].

In our study, we will explain the importance of abstracting to an appropriate level the domain in which the system operates. We will discuss our perspective regarding the process of critiquing therapy, and describe the critiquing model that we developed. To validate our ideas, we created the system HyperCritic. We shall provide details concerning the computational implementation of HyperCritic, and shall examine examples of the system's output. In the description of the model and of its implementation, we shall contrast our approach with those taken by developers of other systems that provide decision support based on automated medical records.

1.3 TASK-BASED MODEL

An architecture for a medical decision-support system that concentrates on modeling the application tasks to be performed generally is referred to as a *task-based architecture*. Several researchers have noted the difference between the procedural aspects of a task and the specific knowledge required to execute that task [13-20]. The unifying theme in these research projects is the notion that the procedural aspect of a given task should be represented separately from the specific knowledge required to execute that task. Different researchers, however, have addressed the issue of separating these knowledge components from different perspectives. Consequently, the knowledge components that they have identified differ and the systems that they have developed illustrate different advantages that can be gained from separating these knowledge components.

The representation of the problem-solving behavior of NEOMYCIN, for instance, on the level of tasks greatly enhanced the explanation facilities of that system because it provided explanations in terms of the tasks that need to be performed in that domain [15,19]. Musen showed in the PROTÉGÉ system how the separate modeling of a task's process and content components can be used to develop knowledge-acquisition tools [17]. The developers of the Oxford System of Medicine showed how the same medical knowledge can be used for a variety of tasks [13,20].

The work presented in this study describes the application of task-based architectures to the problem of critiquing based on automated medical records. In the domain of critiquing systems, the advantages of task-based architectures have not been explored. We propose a task-based model for critiquing physicians' management of patients based on data from automated medical records. In the HyperCritic program we explore the advantages of the resulting task-based architecture for a critiquing system in the domain of the therapeutic management of hypertensive patients.

1.4 EVALUATING THE UNDERLYING ASSUMPTIONS

The underlying assumption of the work presented in this thesis is that data obtained from automated medical records can be used to generate a medically relevant critique. Using medical data collected for one purpose in the context

relationships used in the system. In the case of a relational database, the relational model is expressed in terms of tables in the database and of the operations (for example, select, project, and join) that manipulate the data stored in these tables. In contrast to the explicit data models that undergird database systems, the domain models underlying decision-support systems often are implicit. The behavior of such systems is described in terms of the symbols (for example, rules, frames, or objects) and the inference strategies that manipulate those symbols (for example, forward chaining, backward chaining, or belief update in causal probabilistic networks) [10].

An important perspective articulated by Newell [11] is that knowledge is an abstraction that can be separated from the symbols that are used to represent the knowledge. Knowledge, in the view of Newell, is a set of goals and the behavior potentially needed to achieve those goals. Knowledge itself can never be written down; it can only be observed as an activity. This distinction between knowledge (at what Newell refers to as the *knowledge level*) and the symbols used to represent knowledge (the *symbol level*) allows us to distinguish our goals for an intelligent system from the language that we use at the symbol level to represent these goals. Thus, knowledge-level analysis of an application task specifies the behaviors that are required to solve a problem in the world; analysis of a knowledge base at the symbol level specifies the computational mechanisms needed to model the requisite behavior. Researchers in artificial intelligence (AI) increasingly agree that it is important to understand a domain task in terms of its knowledge-level specifications before proceeding to a symbol-level implementation. Although there is little consensus about how to go about describing domain tasks at the knowledge level, the goal becomes to understand a system's behavior in terms of an abstract model, rather than by means of a specific set of notations [11,12].

In our study, we will explain the importance of abstracting to an appropriate level the domain in which the system operates. We will discuss our perspective regarding the process of critiquing therapy, and describe the critiquing model that we developed. To validate our ideas, we created the system HyperCritic. We shall provide details concerning the computational implementation of HyperCritic, and shall examine examples of the system's output. In the description of the model and of its implementation, we shall contrast our approach with those taken by developers of other systems that provide decision support based on automated medical records.

1.3 TASK-BASED MODEL

An architecture for a medical decision-support system that concentrates on modeling the application tasks to be performed generally is referred to as a *task-based architecture*. Several researchers have noted the difference between the procedural aspects of a task and the specific knowledge required to execute that task [13-20]. The unifying theme in these research projects is the notion that the procedural aspect of a given task should be represented separately from the specific knowledge required to execute that task. Different researchers, however, have addressed the issue of separating these knowledge components from different perspectives. Consequently, the knowledge components that they have identified differ and the systems that they have developed illustrate different advantages that can be gained from separating these knowledge components.

The representation of the problem-solving behavior of NEOMYCIN, for instance, on the level of tasks greatly enhanced the explanation facilities of that system because it provided explanations in terms of the tasks that need to be performed in that domain [15,19]. Musen showed in the PROTÉGÉ system how the separate modeling of a task's process and content components can be used to develop knowledge-acquisition tools [17]. The developers of the Oxford System of Medicine showed how the same medical knowledge can be used for a variety of tasks [13,20].

The work presented in this study describes the application of task-based architectures to the problem of critiquing based on automated medical records. In the domain of critiquing systems, the advantages of task-based architectures have not been explored. We propose a task-based model for critiquing physicians' management of patients based on data from automated medical records. In the HyperCritic program we explore the advantages of the resulting task-based architecture for a critiquing system in the domain of the therapeutic management of hypertensive patients.

1.4 EVALUATING THE UNDERLYING ASSUMPTIONS

The underlying assumption of the work presented in this thesis is that data obtained from automated medical records can be used to generate a medically relevant critique. Using medical data collected for one purpose in the context

of another purpose, however, needs to be approached with utmost care [21]. For example, when medical data collected for billing purposes are used for epidemiological studies, then the data need to be interpreted very carefully in the context of how the financial consequences of a given diagnosis may have effected the prevalence of that diagnosis [22]. Similarly, computer-stored medical records are not explicitly designed as data-entry modules for critiquing systems. Using data from computer-stored medical records as input for critiquing systems, therefore, requires a careful analysis of whether the data in the computer-stored medical records are suitable for that purpose. Is it realistic to expect that physicians, like pilots flying modern jetplanes, will be continuously monitored based on data from computer-based medical records in order to prevent calamities? Or will computer-based critiquing require physicians to change how medical records are kept?

When people interact with their environment, they form models of themselves and of the environment which they are interacting with. Such internal models are known as *mental models*. These mental models provide predictive and explanatory power for understanding the interaction with the environment [23]. Similarly, the physician forms a mental model of the patient whom he is treating. When another physician is asked to critique this treatment, he has to reconstruct the intentions and reasoning of the treating physician. The critiquing physician has to formulate a model of the treating physician's mental model. Such a model of a mental model is known as a conceptual model [24]. In the medical record, one encounters both data describing the patient's state (e.g., the results of laboratory tests) and data describing the mental model of the treating physician (e.g., a description of treatment goals). The creation of such a conceptual model of the treating physician lies at the heart of a critique: The recognition of the intentions (the treatment objectives) of the physician in combination with the actions undertaken to achieve these treatment objectives [25].

In this study, we will evaluate whether the computer-based medical record of a primary-care information system contains enough information to allow another physician to create a conceptual model of the mental model of the treating physician, and subsequently to generate critique. The rationale is that the ability to reconstruct the reasoning of the general practitioner must be a prerequisite for the development of a critiquing system. We will perform a study in which general practitioners (GPs) are asked to provide the computer-based medical records of five patients with hypertension. A printout of these medical records

will be submitted to an internist who had a recognized interest and experience in the treatment of hypertension. The internist will be asked to comment on the treatment of hypertension as documented in the medical records. Subsequently, the comments of the internist will be submitted to a panel of three GPs; these GPs will be asked to judge the relevance of the comments. Finally, the comments of the internist will be shared with the GP who had treated the patient.

1.5 EVALUATING THE CRITIQUING MODEL

Any model is necessary selective in what it contains. In creating models, the unusual properties of special cases are sacrificed to emphasize those of the general situation. A model, ideally, identifies the most appropriate level of abstraction that will allow a particular task to be performed without introducing so much generality and subsequent rigidity that few actual tasks will fit the model. As soon as the application area does not "match" or "fit" the model, the system builder must either adapt the model or discard the model in its entirety [10]. From the system-builders' viewpoint, it is of utmost importance to understand the limitations of a given model.

In this study, we will investigate the possibilities and limitations of our critiquing model, using data from computer-based medical records, by comparing the performance of human observers with the performance of our computer-based critiquing system. We will select a number of patients from several 'paperless' practices (that is, practices in which the physician no longer maintains paper-based medical records). The computer-stored medical records will be submitted to physicians and to the critiquing system. A major restriction that we will impose is that both critiquing system and physicians have to rely solely on the computer-stored medical record; direct interaction with the physician who treated the patients will not be allowed. Subsequently, the comments of the critiquing physicians will be compared with the comments that the critiquing system generates. The purpose of the study is (a) to investigate whether the computer-based medical records contain sufficient information to generate critique by submitting these medical records to both physicians and a critiquing system, and (b) to investigate the limitations of computer-based critiquing compared with the performance of physicians.

1.6 A GUIDE TO THE READER

This study is not written in the form of a monograph, but rather consists of a number of separate papers. This provides the reader with the opportunity to read one or more chapters without necessarily having to read the other chapters. The flip side, however, is that a certain degree of redundancy cannot be avoided. Moreover, the reader who has the energy to read the entire report will discover that the same issue may be discussed from different perspectives in different chapters.

In **Chapter 2**, we will discuss the importance of abstracting to an appropriate level the domain in which a computer-based decision-support system functions. Developers of computer-based decision-support tools frequently adopt either pattern recognition or artificial-intelligence techniques as the basis for their programs. Because these developers often choose to accentuate the differences between these alternative approaches, the more fundamental similarities are frequently overlooked. We argue that the principal challenge in the creation of any clinical consultation program -- regardless of the methodology that is used -- lies in creating a computational model of the application domain. The difficulty in generating such a model manifests itself in symptoms that workers in the expert-systems community have labeled "the knowledge-acquisition bottleneck" and "the problem of brittleness." In this chapter, we explore these two symptoms, and show how the development of consultation programs based on pattern-recognition techniques is subject to analogous difficulties.

In **Chapter 3**, we report the results of study in which we investigate whether the computer-based medical record of a primary-care information system contains enough information to allow a *human* observer to generate critique. In this study we ask both the physician who treated the patient and other physicians to judge the relevance of the critique. We will investigate the limitations of the computer-based medical record, and we will discuss why physicians may judge a comment irrelevant.

In **Chapter 4**, we describe our critiquing model. We view critiquing based on automated medical records as an interpretation of the medical record in order to detect the physician's actions and decisions, followed by the invocation of a limited set of critiquing tasks. These tasks are designated to detect conflicts between the inferred condition of the patient and the recorded decisions the

physician has made. The structure of these critiquing tasks can be separated from the actual medical knowledge required to execute those tasks.

To validate our ideas, we developed a system, called HyperCritic, that is able to critique the decision making of general practitioners caring for patients with hypertension. In Chapter 4, we provide a description of HyperCritic, together with examples of its output[1]. HyperCritic uses the notion of abstract critiquing tasks to structure the medical knowledge encoded in the system. We illustrate the advantages of these additional levels of abstraction in two areas: knowledge acquisition and knowledge maintenance.

In **Chapter 5**, we present the results of a study in which we compare the performance of HyperCritic with the performance of physicians. We will discuss the limitations of computer-based reviewing from three perspectives: (a) limitations due to the available data in the computer-stored medical record, (b) limitations due to the available medical knowledge, and (c) limitations due to HyperCritic.

In **Chapter 6**, we summarize this research.

REFERENCES

[1] Barnett GO, Cimino JJ, Hupp JA, et al. DXplain: An evolving diagnosis decision-support system. *JAMA* 1987;258:67-74.

[2] Miller RA, Masarie FE. The demise of the Greek oracle model for medical diagnosis systems. *Meth Inform Med* 1990;29:1-2.

[3] Shortliffe EH. Computer programs to support medical decision making. *JAMA* 1987;258:61-66.

[4] Miller PL. *Expert Critiquing Systems, Practise-Based Medical Consultation by Computer.* New York: Springer-Verlag, 1986.

[5] McDonald CJ, Hui SL, Smith DM, et al. Reminders to physicians from an introspective computer medical record. *Ann Int Med* 1984;100:130-8.

[6] Pryor TA, Gardner RM, Clayton PD, et al. The HELP system. *J Med Syst* 1983;7:87-102.

[7] Warner HR. *Computer-Assisted Medical Decision Making.* New York: Academic Press, 1978.

[1] Additional technical documentation of programs described in this thesis is available upon request from the Department of Medical Informatics, Erasmus University, Rotterdam, the Netherlands.

[8] Evans RS, Larsen RA, Burke JP, et al. Computer surveillance of hospital-acquired infections and antibiotic use. *JAMA* 1986;256:1007-11.

[9] Langlotz CP, Shortliffe EH. Adapting a consultation system to critique user plans. *Int J Man-Mach Stud* 1983;19:479-96.

[10] Musen MA, Van der Lei J. Knowledge engineering for clinical consultation programs: Modeling the application area. *Meth Inform Med* 1989;28:28-35.

[11] Newell A. The knowledge level. *Artif Intell* 1982;18:87-127.

[12] Clancey WJ. Heuristic classification. *Artif Intell* 1985;27:289-350.

[13] Fox J. Symbolic decision procedures for knowledge based systems. In: Adell H, ed. *The Handbook of Knowledge Engineering.* New York: McGraw-Hill, 1989.

[14] Gruber TR. Acquiring strategic knowledge from experts. *Int J Man-Mach Stud* 1988;29:579-97.

[15] Hasling DW, Clancey WJ, Rennels GD. Strategic explanations for a diagnostic consultation system. In: Coombs MJ, ed. *Developments in Expert Systems.* New York: Academic Press, 1984:117-33.

[16] Lanzola G, Stefanelli M, Barosi G, et al. A knowledge system architecture for diagnostic reasoning. In: Proceedings of the *Second European Conference on Artificial Intelligence in Medicine.* London: Springer Verlag, 1989:234-47.

[17] Musen MA. Automated support for building and extending expert models. *Machine Learning* 1989;4:349-77.

[18] Swartout WR. *Producing Explanations and Justifications of Expert Consulting programs [Dissertation].* Cambridge MA: Massachusetts Institute of Technology, MIT/LCS/TR-251, 1981.

[19] Clancey WJ, Letsinger R. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In: Proceedings of the *Seventh International Joint Conference on Artificial Intelligence.* Vancouver, Canada, 1981:829-36.

[20] Glowinski A, O'Neil M, Fox J. Design of a generic information system and its application to primary care. In: Proceedings of the *Second European Conference on Artificial Intelligence in Medicine.* London: Springer Verlag, 1989:221-33.

[21] Musen MA. The strained quality of medical data. *Method Inform Med* 1989;28:123-5.

[22] Burnum JF. The misinformation era: The fall of the medical record. *Ann Int Med* 1989;110:482-4.

[23] Norman DA. Some observations on mental models. In: Gertner D, Stevens AL, eds. *Mental Models.* Hillsdale NJ: Lawrence Erlbaum, 1983:7-14.

[24] Gertner D, Stevens AL, eds. *Mental Models.* Hillsdale NJ: Lawrence Erlbaum, 1983.

[25] Miller PL. Goal-directed critiquing by computer: ventilator management. *Comp Biomed Res* 1985;18:422-38.

CHAPTER 2

Knowledge Engineering for Clinical Consultation Programs: Modeling the Application Area

Published in Methods of Information in Medicine
1989;28:28-35

Mark A. Musen
Johan van der Lei

ABSTRACT

Developers of computer-based decision-support tools frequently adopt either pattern recognition or artificial-intelligence techniques as the basis for their programs. Because these developers often choose to accentuate the differences between these alternative approaches, the more fundamental similarities are frequently overlooked. The principal challenge in the creation of any clinical consultation program -- regardless of the methodology that is used -- lies in creating a computational model of the application domain. The difficulty in generating such a model manifests itself in symptoms that workers in the expert-systems community have labeled "the knowledge-acquisition bottleneck" and "the problem of brittleness." This paper explores these two symptoms, and shows how the development of consultation programs based on pattern-recognition techniques is subject to analogous difficulties. The expert-systems and pattern-recognition communities must recognize that they face similar challenges, and must unite to develop methods that assist with the process of building of models of complex application tasks.

Key Words : Decision-Support Systems, Expert Systems, Pattern Recognition, Knowledge Acquisition, Domain Modeling

2.1 INTRODUCTION

Workers in medical informatics have experimented with a variety of computer-based approaches to assist with medical decision making [1]. Many researchers have developed clinically useful programs using branching logic and various statistical pattern-recognition methods. Recently, artificial intelligence (AI) techniques have fostered the creation of medical expert systems that can solve problems in ways that are understandable to physicians and that can explain the basis of the programs' recommendations in intuitive ways [2, 3]. In the past, discussions of these alternative paradigms have frequently concentrated on the different assumptions that are inherent in the methodologies. Builders of expert systems and of statistical pattern-recognition systems have had intense debates on the applicability and usefulness of their respective techniques. This emphasis on the differences among approaches has often obscured more fundamental similarities.

Because much current work on the development of computer-based medical decision aids centers on the use of either AI or pattern-recognition methods (including Bayesian statistics), our discussion in this paper is limited to these two disciplines. We examine the two factors that workers in AI repeatedly identify as the principal barriers to the widespread deployment of expert systems, and show how these factors pose equal challenges to workers in the pattern-recognition community -- and, ultimately, to the builders of all decision-support systems. In the jargon of AI, these universally-recognized barriers to the dissemination of expert systems are (1) the knowledge-acquisition bottleneck and (2) the problem of "brittleness" [4]. The process of knowledge acquisition traditionally concerns the elicitation and encoding of a given professional's relevant expertise to create the knowledge base of an expert system. Brittleness refers to the failure of an expert system to offer appropriate advice on classes of cases that the developers may not have considered at the time that they created the expert system. These expressions represent more than just AI buzzwords; they denote major concerns that scores of researchers in the expert-system community are actively attempting to address (for example, see [5,6]). Brittle system behavior and difficulties in acquiring and maintaining knowledge are not, of course, the only impediments to the dissemination of clinical decision-support programs [7]. We concentrate on these issues because they are so fundamental, and because they are of central concern to workers in medical informatics.

The knowledge-acquisition bottleneck and the problem of brittleness are symptoms of the underlying difficulty that developers invariably face when they attempt to construct a computational model of a given expert-system application area (domain). Although the particular terminology may be unique to AI, the problem of domain modeling also extends to other disciplines. This paper examines the related problems of knowledge acquisition and brittleness in the construction of clinical decision-support programs. Medical expert systems are seen to incorporate non-numeric, qualitative models of human behavior [8], whereas the models in statistical classifiers reflect inherently stochastic associations[1]. By elucidating the aspects of domain modeling that are common to both approaches, this paper offers a new perspective from which expert and pattern-recognition systems can be analyzed and compared. Although not emphasized in this paper, the analysis also applies to algorithmic approaches, normative decision theory, and other techniques that may be used in the construction of medical decision-support tools.

2.2 THE KNOWLEDGE-ACQUISITION BOTTLENECK

The knowledge bases of expert systems typically are built by specially trained programmers called knowledge engineers. Knowledge engineers interview application specialists (domain experts) and attempt to identify how those experts make professional decisions. The engineers then encode the knowledge that they elicit using a special-purpose representation, such as if/then production rules. The expression "transfer of expertise" pervades the AI literature, indicating that expert-system builders frequently view knowledge acquisition as a problem in the transfer of knowledge from the minds of domain experts to the knowledge bases of expert systems to, ultimately, the expert-system users. The metaphor of a bottleneck that impedes this transfer is in many ways quite appropriate, as prodigious amounts of time and human resources are required to build and refine expert-system knowledge bases. For example, development of the commercial expert systems that are now marketed by AI start-up companies typically required between 20 and 50 person-years [9].

The laborious nature of knowledge acquisition is commonly ascribed to

[1]Syntactic pattern recognition programs have models with both qualitative and stochastic elements; to ease our discussion, however, we shall concentrate only on purely qualitative or purely stochastic systems.

problems in communication [10,11]. Knowledge engineers who are unfamiliar with a system's intended application area may not always grasp the significance of what a domain expert relates to them, and, problematically, frequently do not even know the questions that they should ask in the first place. At the same time, experts in the application area may have little appreciation of how knowledge bases are constructed and may have no idea what aspects of their expertise need to be encoded. The many cycles of system building, testing, and rebuilding that are common in the AI industry are thus widely attributed to the inability of knowledge engineers and application specialists to speak the same language. Recently, however, researchers have begun to recognize that the knowledge-acquisition bottleneck reflects more than just a problem in transferring expertise through a channel of narrow bandwidth. The difficulty may lie not so much with the bottle as with its contents.

2.2.1 Knowledge Acquisition and the Nature of Expertise

Knowledge acquisition is difficult because experts often do not themselves know how they classify objects and solve problems. Although application specialists may readily describe the categories into which they sort the entities that they encounter in their profession and may willingly delineate the features that would seem to form the basis of their decision making, there is no guarantee that such introspective reports are reliable [12,13]. When knowledge engineers question experts about activities that the experts perform routinely without much conscious thought, the experts frequently offer plausible answers that may not accurately reflect their true behavior. For example, Slovic and Lictenstein [14] asked stock brokers to weight the importance of various features of investments that seemed to affect the brokers' trading decisions. A regression analysis of actual decisions made by the stock brokers revealed computed weights for these factors that correlated poorly with the brokers' subjective ratings. In another well-described example, Michalski and Chilausky [15] found that decision rules elicited from plant pathologists for the classification of soybean diseases performed less accurately than did a different rule set that was induced by computer from a library of test cases that the plant pathologists had previously diagnosed.

In both these cases, the domain experts were neither being capricious nor intentionally trying to mislead the investigators; the subjects of the experiments had attempted to describe their professional knowledge as well as they could

and had communicated successfully those descriptions to the investigators. The dilemma arose in each situation because (1) the domain experts -- like all human beings -- lacked the ability for reliable introspection about the highly skilled knowledge that they used during proficient problem solving [12,13], and (2) the subjects -- like everyone in Western society -- were immersed in a culture that had taught them, paradoxically, that such accurate introspection is somehow possible [16]. Although people may believe that they have insight into their skilled reasoning, cognitive psychologists have proved that such intuitions are often incorrect. The stock brokers and plant pathologists thus expounded on their particular decision-making behavior with self-confidence, but explained their actions in terms of decision rules that were later shown to be inferior to the subjects' native thinking.

The build-test-rebuild cycles that make knowledge acquisition so laborious are thus not simply the result of misunderstandings between application specialists and knowledge engineers. Much of the difficulty stems from the inability of experts to know how they actually solve problems in their professions [12,17]. Knowledge acquisition consequently requires that knowledge engineers and application specialists work together to formulate original models of problem solving -- models that can achieve expert-level performance when implemented as computer programs. Knowledge engineers generally strive to create, as far as possible, computational models that approximate the experts' observed behavior. Consequently, construction of such models is bottlenecked not simply because domain experts and knowledge engineers miscommunicate, but primarily because the experts have so little insight into their own expertise.

2.2.2 Creating Classification Models

Most expert systems perform classification. Such systems use the values of features that are associated with entities in the world to determine abstract classes that describe those entities. Then, on the basis of the abstract classes, the expert systems offer recommendations to their users[2]. The MYCIN program

[2]This paper concentrates on AI methods that solve problems by classification, selecting a solution from a pre-enumerated set. Other AI methods construct solutions to problems, and are applied to those tasks for which the set of possible solutions cannot be determined in advance. An expert system that created a new therapy plan, for example, might require such a constructive method [32]. We have omitted the subject of constructive methods from our discussion because such methods generally are applied to tasks for which pattern-recognition techniques are not suitable. The domain-modeling problems that we describe for expert systems that perform classification, however, pertain equally to constructive expert systems.

[18], for instance, requires its users to enter patient-specific feature values (for example, that a patient's white-blood-cell count is 2000 per cubic millimeter) that the program uses to generate appropriate abstractions (for example, that the patient has leukopenia, and, consequently, that the patient is a "compromised host"). The abstract classifications, in turn, suggest MYCIN's recommendations (for example, that the patient may be infected with the bacterium Escherichia coli, and, therefore, that a particular antibiotic is indicated).

Clancey's model of heuristic classification [19] provides a useful set of terms and relationships that can describe not only the behavior of MYCIN, but also that of a large number of other expert systems (Figure 1). The heuristic-classification model allows knowledge acquisition for such expert systems to be viewed as a matter of defining (1) entities in the world to be classified (for example, patients with possible meningitis), (2) relevant features of those entities (for example, white-blood-cell count), (3) abstractions of those features (for example, leukopenia), (4) recommendations that the expert system might suggest (for example, "treat for E. coli"), and (5) heuristics that link classifications to appropriate recommendations (for example, that patients classified as "compromised hosts" are likely to be infected with a class of bacteria of which E. coli is a member). The heuristic-classification model thus provides a framework within which knowledge engineers and application specialists can structure their thoughts about an expert system's reasoning.

The availability of such a framework is clearly helpful, but enormous problems still remain for the builders of expert systems. What are the relevant features that need to be represented? What are the appropriate abstractions for those features and the rules for performing those abstractions? Knowledge engineers cannot simply expect domain specialists to tell them the answers to questions such as these. Instead, the engineers and the experts must work together to build and test different models of the application task -- models that may involve various sets of features and various types of abstractions. This active and inventive modeling process is the essence of knowledge acquisition. As in the design of other types of software systems [20], the need for creativity makes the development of such models the major bottleneck step in the construction of AI consultation programs.

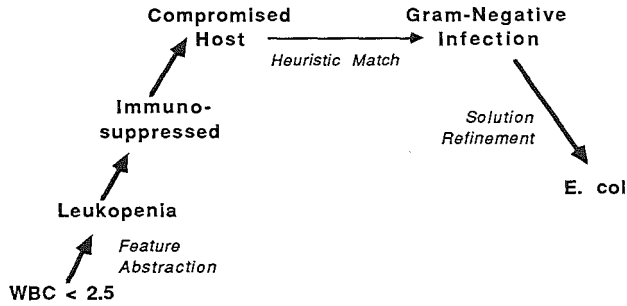

Figure 1:
Heuristic classification is a problem-solving method in which a selection is made from a preenumerated set. Heuristics link feature-abstraction hierarchies to solution-refinement hierarchies. Here, the method is applied to the organism-identification task in MYCIN. WBC stands for white-blood-cell count. (Source: Adapted from Clancey [19]).

2.3 BRITTLENESS

Despite their often impressive performance, expert systems have clear limitations. Because knowledge bases are models, they are necessarily selective in what they contain. MYCIN, for example, incorporates no specific knowledge of many common infections, such as sinusitis or pneumonia. Before allowing a user to enter the data concerning such infections, the program prints out a disclaimer stating that the knowledge base is incomplete and that only cases of bacteremia or meningitis should be entered into the system. The user, of course, can enter data on any patient he chooses. Thus, if the user asks MYCIN to reason about a patient with pneumonia as though the patient had bacteremia, the program will offer a recommendation, but it is unclear how the user should interpret such advice. Even if the user complies with MYCIN's disclaimer and enters only "appropriate" cases, problems still, however, arise. Many of the antibiotics that the program recommends have been supplanted by more effective drugs in the past decade; certain diagnostic possibilities that have been described only recently (for example, Legionnaire's disease and certain infections associated with AIDS) also are absent from the knowledge base. MYCIN does not "know" about these deficiencies in its knowledge. More important, a "naïve" user of the system would have no way of ascertaining the scope of MYCIN's ignorance.

Sometimes, an expert system will be developed that has explicit knowledge of some of its own limitations. The ONCOCIN system [21], for example, advises physicians on the administration of cancer chemotherapy according to predefined treatment plans called protocols. In addition to those production rules that define the administration of cancer-treatment plans, the ONCOCIN knowledge base contains several rules, such as the following, which cause the system to forego making any recommendation at all:

Rule 176

IF (1) The bone-marrow biopsy was positive,
(2) This is not the patient's first visit to the clinic, and
(3) A: The patient's white-blood-cell count is too low for therapy to be given, or
B: The patient's platelet count is too low for therapy to be given,
THEN Conclude that the protocol requires consultation.

Thus, the model of cancer therapy in ONCOCIN makes explicit a number of unusual situations that the program's developers anticipated would be difficult for the expert system to handle. When it encounters such circumstances, the system tells the user to consult a senior cancer specialist, as ONCOCIN itself can offer no suggestion for treatment.

There are, however, myriad special circumstances that system builders never think to incorporate into their initial models of an application area and that cannot be handled by special rules. Shortliffe [7], for example, tells an enlightening anecdote of how a physician once chose to totally disregard ONCOCIN's recommendation that a young woman with Hodgkin's disease receive full doses of chemotherapy. The physician believed that ONCOCIN's decision was incorrect -- not because the patient was experiencing unusual problems with the treatment, but because the patient was to be a bridesmaid at a wedding the following day and did not want to be nauseated from chemotherapy during the ceremony. Weddings and bridesmaids are not part of ONCOCIN's model of cancer therapy, just as AIDS and Legionnaire's disease are not part of MYCIN's model of infectious diseases. When there are relevant features of a case that are outside an expert system's model of the application area, the expert system is likely to fail. Because expert-system knowledge

bases, like all models, are necessarily selective in what they contain, AI systems can appear brittle when confronted with even slightly unusual cases.
Developers of expert systems, while researching ways to construct more complete models, underscore the need for human users to interpret a system's recommendations in light of a common-sense understanding of the application area. Users of expert systems -- like users of any kind of tool -- must learn to recognize the technology's limitations.

2.4 DOMAIN MODELING IN PATTERN RECOGNITION

The qualitative models in expert systems contrast sharply with the quantitative models underlying statistical pattern-recognition approaches. Qualitative expert-system models may incorporate numerous relationships among entities. Such relationships may include composition (for example, that data concerning a patient with a presumed infectious disease should include data about microbiological cultures), taxonomy (that E. coli is a gram-negative bacterium), and causality (that "compromised hosts" are susceptible to infections with gram-negative bacteria). Formal models of problem-solving methods, such as Clancey's description of heuristic classification (see Figure 1), can help knowledge engineers to clarify the semantics of the task models that they build into the knowledge bases of expert systems, and thus can provide some theoretical foundation for work in applied AI. In practice, however, the semantics of expert-system models often involve operational definitions that are established by the pragmatics of a particular application domain, rather than by a pre-existing theory.

In pattern recognition, on the other hand, there are many well-understood statistical principles that relate the features of entities to appropriate classifications. Unlike the development of expert-system models, it is impossible to construct pattern-classification models without paying attention to an underlying theory. Nevertheless, the presence of powerful theories of statistical classification should not obscure the fact that all useful pattern-recognition systems, like all expert systems, are the products of human attempts to model some reality.

The terms and relationships in a pattern-recognition model establish (1) the classes into which entities in the domain will be grouped, and (2) the features of those entities that will discriminate among the different classes. Despite the

fundamental importance of identifying these classes and features before applying any statistical algorithm to a classification task, the literature of the pattern-recognition community has not concentrated on the crucial problem of developing a domain model. In 1969, Kanal and Chandrasekaran lamented,

"For all practical purposes the theory of pattern recognition has become identical to the statistical theory of classification. The reason for this lopsided development is that theorizing on the essentially heuristic job of feature formation is extremely difficult, even if formalizing heuristics is not a contradiction in terms." [22]

Seventeen years later, Nagao observed, "We have no good answer yet for the problem of what kind of features to detect for the recognition of a set of patterns. Pattern features are determined by the instinct of the system designer without any theoretical reasons" [23].

The literature may downplay the problem of domain modeling because of the way developers created many of the first pattern-recognition systems. Frequently, application specialists worked independently to amass large data sets of cases and corresponding feature values, and then presented those data to their colleagues in the pattern-recognition field for construction of appropriate classifiers. In such circumstances, application specialists did the domain modeling a priori, and they themselves decided which features might be relevant and what classifications should be made. The statisticians then applied domain-independent techniques to select subsets of features to include in the final models [24] and to determine algorithms for distinguishing among cases on the basis of the cases' feature vectors. If there was a bottleneck in deciding which feature values to measure in the first place, that stumbling block was often considered to be a problem for the application specialist, not the pattern-recognition expert, to solve.

The more typical scenario that has emerged, of course, is either for the application specialist and the pattern-recognition specialist to work in concert in the creation of an automated classifier, or for the roles of the two specialists to be shared by the same person. In such situations, the development of pattern-recognition systems is seen more clearly as a highly iterative model-building process. The system developers must identify possibly relevant features and must formalize the feature definitions; they must select candidate subsets of those features for inclusion in a statistical model, with no guarantee

that any particular subset will be optimal; they must then sift through and test potential computational algorithms on appropriate training samples. At each step, the builders of the pattern-recognition model may have to backtrack as their insight into the classification problem improves. Although interactive computer-based tools may greatly facilitate this process, many build-test-rebuild cycles are often necessary [25], just as occurs in the construction of expert-system models.

To our knowledge, no one in the pattern-recognition community has studied formally how application specialists first identify the features that they propose for use by statistical classifiers. Data from the psychological literature [13,16] suggests that people's inability for reliable introspection into their own skilled behavior is a major obstacle. As Kanal and Chandrasekaran [22] succinctly stated, "The quality of decisions we make depends on the quality of questions we ask." The untrustworthy nature of human self-reporting makes it unlikely that application specialists ever articulate all the useful features in a domain. Although well-understood theories can help the builders of pattern-recognition systems to discard seemingly irrelevant or redundant features from a classification model [24], workers in pattern recognition have paid little attention to developing techniques to assist domain experts in identifying a set of candidate features in the first place. The difficult nature of the feature-formation problem is probably the root of many of the build-test-rebuild cycles that pervade the construction of statistical classifiers.

2.5 THE ANALOG OF BRITTLENESS

Although the pattern-recognition literature does not mention the word "brittle," statistical classifiers, like expert systems, unquestionably make assumptions about their application domains. Munson described the now well-recognized problem in 1969:

"A system that performs well, for example, on any number of handwriting samples gathered in the designer's laboratory may prove embarrassingly poor when confronted with the writing of the general public on bank checks, envelopes, or credit cards. A designer might work long and hard on an electrocardiogram-classifying or speech-recognizing system, and achieve excellent results using design data and independent test data from a large random sampling of adult males, only to find that his classifying system gets

nowhere with data from children and females -- or from hearts acclimatized to high altitudes or voices from a part of the country not covered in the experimental samples." [26]

Most pattern-recognition systems show diminished performance when they attempt to classify test cases that were not elements of the systems' training sets. Much of this reduction in accuracy is simply a function of random variation in the values of features of the entities being classified -- features that in practice prove to be of lower discriminatory power than the training-set cases would indicate. Pattern-recognition systems may also fail, however, due to problems in their design.

Just like builders of expert systems, developers of pattern-recognition systems cannot avoid having particular types of cases in mind when they propose the features and the classes to be used as the basis for a classification model. Although the model can be refined using domain-independent techniques, it is always dependent on a human being's knowledge of the intended application area, and is likely to be biased by the salient features of available cases. As Munson concludes, "the influence of the particular problem at hand does unavoidably creep into the design of a recognition system in various large and small ways, and . . . the system becomes tailored to the specifics of the problem" [26]. Thus, if the developer of a system to classify white blood cells does not think to incorporate features into the classification model that will allow discrimination of nucleated red blood cells from leukocytes, the system will be "brittle" for smears that contain nucleated red cells; if the developer of a Bayesian diagnostic system such as de Dombal's program for acute abdominal pain [27] does not think to include schistosomiasis as a possible disease entity, then the system will appear brittle whenever it is presented with patients from an area where schistosomiasis is endemic. As with expert systems, the degree of brittleness is a function of the selectivity of the underlying domain model.

Pattern-recognition systems may explicitly identify cases that are not classifiable given the underlying model. For example, a white-blood-cell classifier may determine from a white blood cell's feature vector that the cell does not fall into a known category, and thus may assign the cell to a reject class. In some sense, this behavior indicates that the classifier has some notion of the limitations of its abilities -- much like expert systems such as ONCOCIN, which contains rules to determine when a case has features that make it impossible for the system to offer a satisfactory recommendation. There will always be

features, however, that are outside the domain models of expert systems and pattern-recognition systems -- features such as "heart has been acclimatized to high altitude" or "patient wants to be a bridesmaid tomorrow." When such features are relevant in classifying a particular case, our systems will fail. More important, because such failures are due to features that are not part of the underlying model, our systems will perform whatever classifications a user requests and arrive at possibly incorrect results without giving any indication that an aberrant case is unusual or atypical. It is therefore up to the user, whose domain model is presumably more encompassing than that of the classification program, to identify when the behavior of an expert system -- or that of a pattern-recognition system -- is brittle.

2.6 DISCUSSION

The current AI literature repeatedly acknowledges that expert systems are both laborious to build and are vulnerable to poor performance when confronted with unusual cases. Whereas the notions of a knowledge-acquisition bottleneck and of brittleness apply equally to pattern-recognition systems, only the very early pattern-recognition literature seems to emphasize these concerns in a comparable manner. What is viewed as a major problem for the expert-systems community is tacitly acknowledged, although infrequently discussed, among workers in pattern recognition.

The difference in attitudes toward these problems in system development may stem from differences in both methodology and research goals. Although the two communities share the same obstacles in creating and validating domain models, the explicit structure imposed on those domain models by workers in the pattern-recognition field, and the scope of the models attempted by many developers of expert systems, account for much of the disparity.

2.6.1 The Elucidation of Domain Models

The classification models used by pattern-recognition systems are, in many ways, quite powerful. The models' performance can be understood in terms of well-defined theories. The assumptions that the models make about the classification process are explicit. Thus, pattern-recognition models can provide system builders with the unique ability to discard seemingly irrelevant or

redundant features, based on specific knowledge of how those features might contribute to the classification task in the context of the chosen statistical method. The clarity of pattern-recognition models also allows workers to make good estimates of the sizes required for the training and testing populations, based on the number of features in the model and the type of classification algorithm. The remarkable precision with which workers in the pattern-recognition community can view the classification process led Nagao to assert that "we have reached the stage that we can judge fairly easily whether a given pattern recognition problem can be solvable by the present-day pattern recognition technology" [23].

In the expert-systems field, on the other hand, knowledge engineers often require periods of substantial trial and error before they can establish whether a given task can be suitably encoded within a knowledge base. AI programmers initially may lack a clear understanding of either the task that a proposed expert system is intended to solve or the methods by which the system might accomplish that task. Nevertheless, it is still possible to begin exploratory programming of the expert system, even though the model is ill formed [28]. (Whether it is advisable to program in this manner is another matter.) Once knowledge engineers finally build an expert system, there are few formal guidelines that can help them to validate the system's performance [29,30]. Expert systems generally lack underlying theories that can direct the evaluation process.

The AI community has made great progress in recent years in its search for more powerful theories with which to understand the behavior of expert systems. Clancey's description of heuristic-classification problem solving [19] was an important advance that made it possible to describe the actions of programs such as MYCIN independent of the way in which those programs were implemented. The heuristic-classification model provided workers in AI with a set of terms by which MYCIN and a host of other expert systems could be compared -- not on the basis of the arcane symbols that knowledge engineers had used to encode domain-specific knowledge, but on the basis of the systems' behaviors. When knowledge engineers descend to the symbol level and describe their systems in terms of production rules, frames, and LISP code, it is impossible to understand precisely what the systems attempt to model; stating that an expert system uses "forward-chaining production rules" or "frame hierarchies" is no more illuminating than stating that a linear-discriminant function uses "sequences of arithmetic operations."

Recently, the identification of the heuristic-classification method and other abstract problem-solving strategies [31,32] has begun to provide a vocabulary with which workers in AI can describe the qualitative models that expert-system knowledge bases represent. Placing emphasis on the models, rather than on the implementations of those models, has begun to facilitate communication among workers throughout the expert-system community. Moreover, now that the builders of expert systems can describe their classification models explicitly, it has become possible to compare expert-system models with pattern-recognition models more directly. Consequently, members of the two research communities can more fully appreciate how expert-systems techniques and pattern-recognition approaches complement one another.

2.6.2 Overcoming Brittleness

For developers using AI methods, this recent emphasis on the models underlying expert-system knowledge bases provides a perspective that allows knowledge engineers to reinforce their systems against the problems of brittleness. Computer-based knowledge-acquisition tools now allow system builders to enter domain knowledge in terms of explicit problem-solving models such as heuristic classification; the particular production rules or frames that might be used to implement the models thus become secondary. Such tools can critique a user's model in important ways, pointing out areas of the model where the features may not adequately distinguish among competing hypotheses or where frank contradictions occur [32]. Other experimental tools can display elements of a user's model graphically, making the structure and the assumptions of the model more vivid by using visual metaphors in the display of crucial terms and relationships [33]. These novel approaches, however, still require the knowledge engineer to develop an initial model of the application domain. There is no computer-based tool that can detect the brittleness that stems from the inability of application experts to articulate all the relevant features in a given classification task. Similarly, no tool can detect those situations in which system builders inadvertently construct an overly restrictive domain model by failing to consider unusual cases. Domain modeling will always present challenges for the builders of expert systems, as well as for the builders of pattern-recognition systems.

Unlike expert-system models, the classification models used in pattern

recognition are well defined. Workers in the pattern-recognition community build models that are based on explicit statistical theories. Moreover, these workers often strive to build the most parsimonious models possible -- eliminating both redundant features and irrelevant class assignments. Statistical pattern-recognition models thus have a certain "elegance" that is generally missing from the qualitative models in AI. Because the statistical relationships in the models are well understood, and because the terms in the model are minimized, the strong assumptions that pattern-recognition models make about the process of classification are always conspicuous. The relationships between possible feature values and potential classifications are never ambiguous. Thus, a major source of brittleness that occurs in expert systems that incorporate less explicit classification models is obviated. It is likely that the notion of brittleness has not received comparable attention within the pattern-recognition community primarily because of the clarity of pattern-recognition models.

2.6.3 The Challenge of Modeling

Despite the explicit classification models used in pattern-recognition systems and, increasingly, in expert systems, the difficulties of domain modeling remain. Kanal and Chandrasekaran [22] emphasized in the 1960s that the first step in the generation of a pattern-recognition system was for the developers to redefine a perception-recognition task as a classification task. Powerful, transparent methods of classification do nothing to assure the validity of such a redefinition. In fact, attempts to pigeonhole a particular domain task into a particular classification method may either cause system builders to model the task by making significant (brittle) assumptions about the perception-recognition problem, or may cause the developers to reject the domain task from consideration because it is "inappropriate." When workers in the pattern-recognition field describe their discipline as "a bag of tools for a bag of problems" [25], they emphasize their desire to concentrate on a set of well-understood methods and to selectively apply those methods to domain tasks.

Workers in the AI community, on the other hand, have traditionally approached their work from a different perspective. Expert-system builders have been inclined to accentuate the task knowledge that they encode within their knowledge bases, de-emphasizing the problem-solving strategies required to achieve intelligent behavior. Knowledge engineers have not viewed their work

as the application of a "bag of tools"; in fact, an inclination to view each new task as requiring its own ad hoc solution strategy has obscured the similarities among expert-system models and has impeded the identification of general principles of knowledge engineering.

Because of their tendency to subordinate the importance of general problem-solving methods, many expert-system builders have been undaunted in tackling diverse application tasks -- even tasks for which domain experts themselves have no known solution strategies (for example, see [34]). Although this adventurousness of the AI community has led to the development of large numbers of impressive consultation programs, a high price has often been paid. Knowledge engineers have been required to build knowledge bases in the absence of formal theories for knowledge-base validation, allowing end users to stumble upon anomalous, brittle behavior when the expert systems are deployed. At the same time, knowledge acquisition has been bottlenecked by the absence of explicit theories that define how task-specific knowledge is used during problem-solving, making empiric observation of the behavior of completed systems the only well-established means to assess new knowledge bases for internal consistency and completeness. Nevertheless, as the AI community continues to identify and to apply generic problem-solving models (such as that of heuristic classification), both expert-system validation and knowledge acquisition will establish firmer theoretical foundations. As workers in applied AI concentrate more on their modeling activities and less on the symbols with which they encode their models, expert systems should begin to accrue much of the precision and predictability in behavior that are widely associated with pattern-recognition systems.

The problems of knowledge acquisition and of system validation will not, however, be easily solved by either research community, however. The qualitative models used in expert systems and the statistical models used in pattern recognition have inherent limitations. These limitations result not from the particular methodologies used in the two disciplines, but rather from the human source of the models. Constructing classification models is inherently difficult; even the most skilled and the most articulate expert cannot be expected to identify in advance all the features and all the classes that may be relevant to a given classification problem. Expert and pattern-recognition systems will always be based on the imperfect models of the world that are fashioned by the programs' human creators. Researchers in both fields must be cognizant of these limitations, and must work to develop appropriate

strategies to aid system builders in the design of better models. The two communities must recognize the common goals and the common challenges that confront them, and work together toward greater integration of methods for constructing clinically useful systems.

ACKNOWLEDGMENTS

This work was supported in part by grant NF-65/62-521 from the Netherlands Organization for Scientific Research (NWO). Dr. E.S. Gelsema, Dr. T. Timmers, and Dr. J.H. van Bemmel offered valuable comments on a previous draft of this paper.

REFERENCES

[1] Shortliffe EH, Buchanan, BG, Feigenbaum EA. Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. *Proc IEEE 1979*;67:1207-24.

[2] Clancey WJ, Shortliffe EH. *Readings in Medical Artificial Intelligence: The First Decade.* Reading Mass: Addison-Wesley, 1984.

[3] Szolovits P, Patil RS, Schwartz WB. Artificial intelligence in medical diagnosis. *Ann Intern Med* 1988;108:80-87.

[4] Duda RO, Shortliffe EH. Expert systems research. *Science* 1983;220:261-8.

[5] Holland JH. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In: Michalski RS, Carbonell JG, Mitchell TM, eds. *Machine Learning: An Artificial Intelligence Approach II.* Los Altos CA: Morgan-Kaufman, 1986:593-623.

[6] Lenat D, Prakash M, Shepherd M. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine* 1986;6:65-85.

[7] Shortliffe EH. Computer programs to support medical decision making. *JAMA* 1987;258:61-6.

[8] Clancey WJ. *Viewing Knowledge Bases as Qualitative Models.* Stanford CA: Knowledge Systems Laboratory, Stanford University, Technical Report KSL-86-27, 1986.

[9] Spang S, ed. The new AI pioneers: The knowledge merchants. *Spang Robinson Report on AI* 1987;3:1-8.

[10] Buchanan BG, Barstow D, Bechtal R, et al. Constructing an expert system. In: Hayes-Roth F, Waterman DA, Lenat DB, eds. *Building Expert Systems.* Reading Mass: Addison-Wesley, 1983:127-67.

[11] Feigenbaum EA. Knowledge engineering: The applied side of artificial intelligence. *Ann New York Acad Sci* 1984;246:91-107.

[12] Johnson PE. What kind of expert should a system be? *J Med Philos* 1983;8:77-97.

[13] Nisbett RE, Wilson TD. Telling more than we can know: Verbal reports on mental processes. *Psychol Rev* 1977;84:231-59.

[14] Slovic P, Lictenstein S. Comparison of Bayesian and regression approaches to the study of information processing in judgment. *Organiz Behav Human Perf* 1971;6:649-744.

[15] Michalski RS, Chilausky RL. Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology. Internat *J Man-Machine Stud* 1980;12:63-87.

[16] Lyons W. *The Disappearance of Introspection.* Cambridge Mass: MIT Press, 1986.

[17] Winograd T, Flores F. *Understanding Computers and Cognition:A New Foundation for Design.* Norwood NJ: Ablex, 1986.

[18] Buchanan BG, Shortliffe EH. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading Mass: Addison-Wesley, 1984.

[19] Clancey WJ. Heuristic classification. *Artif Intell* 1985;27:289-350.

[20] Brooks FP. No silver bullet: Essence and accidents of software engineering. *Computer* 1987;20:10-19.

[21] Shortliffe EH, Scott AC, Bischoff MB et al. ONCOCIN: An expert system for oncology protocol management. In: *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.* Vancouver BC 1981:876-81.

[22] Kanal LN, Chandrasekaran B. Recognition, machine "recognition," and statistical approaches. In: Watanabe S, ed. *Methodologies of Pattern Recognition.* New York: Academic Press, 1969;317-32.

[23] Nagao M. Toward a flexible pattern analysis method. In: *Proceedings of the Eighth International Joint Conference on Pattern Recognition.* Paris: IEEE Computer Society Press, 1986;170-4.

[24] Stearns SD. On selecting features for pattern classifiers. In: *Proceedings of the Third International Joint Conference on Pattern Recognition.* Coronado CA: IEEE Computer Society Press 1976;71-5.

[25] Kanal LN. Interactive pattern analysis and classification systems: A survey and commentary. *Proc IEEE* 1972;60:1200-15.

[26] Munson JH. Some views on pattern-recognition methodology. In: Watanabe S, ed. *Methodologies of Pattern Recognition.* New York: Academic Press, 1969;417-36.

[27] De Dombal FT, Leaper DJ, Staniland JR, et al. Computer aided diagnosis of acute abdominal pain. *Br Med J* 1972;2:9-13.

[28] Doyle J. Expert systems and the "myth" of symbolic programming. *IEEE Trans Softw Eng* 1985;SE–11:1386-90.

[29] Gaschnig J, Klahr P, Pople H, et al. Evaluation of expert systems: Issues and case studies. In: Hayes-Roth F, Waterman DA, Lenat DB, eds. *Building Expert Systems.* Reading Mass. Addison-Wesley, 1983:241-80.

[30] Miller PM. Evaluation of artificial intelligence systems in medicine. In: Miller PM, ed. *Selected Topics in Medical Artificial Intelligence.* New York: Springer Verlag, 1988;202-11.

[31] Chandrasekaran B. From numbers to symbols to knowledge structures: Pattern recognition and artificial intelligence perspectives of the classification task. In: Gelsema ES, Kanal LN, eds. *Pattern Recognition in Practice II.* Amsterdam: Elsevier, 1986:547-59.

[32] McDermott J. Making expert systems explicit. In: *Information Processing 86.* Dublin: IFIP, 1986;539-44.

[33] Musen MA. *Generation of Model-Based Knowledge-Acquisition Tools for Clinical-Trial Advice Systems.* Ph.D. Thesis, Stanford University CA, Technical Report STAN-CS-88-1194, 1988.

[34] Altman RB, Jardetzky O. New strategies for the determination of macromolecular structure in solution. *J Biochem* 1986;100:1403-23.

CHAPTER 3

Critiquing Expert Critiques.
Issues for the Development of Computer-Based Monitoring in Primary Care

Published in Proceedings of the Sixth World Conference on
Medical Informatics. Amsterdam: Noth-Holland Publ Comp, 1989:106-10

Johan van der Lei, Paul van der Heijden, Wilfried M. Boon

ABSTRACT

A number of workers in Artificial Intelligence have argued that, for some medical domains, critiquing the users decisions is an appropriate approach. Others have stressed the importance of integrating decision-support systems with existing information systems. Little research, however, has been directed to the issues of what consists a relevant critique and whether such a critique could be generated using data obtained from automated medical records. We therefore performed a study in which a general practitioner (GP) was asked to provide us with the computer-based medical records of five patients with hypertension. A printout of these medical records were submitted to an internist who had a recognised interest and experience in the treatment of hypertension. The internist was asked to comment on the treatment of the hypertension as documented in the medical records. Subsequently the comments of the internist were submitted to a panel of three GPs; these GPs were asked to judge the relevance of the comments. Finally the comments of the internist were shared with the GP who had treated the patient.

The internist generated 48 comments. When the GPs were asked to judge the comments of the internist, over 50 percent of these comments were judged relevant -- but there was little consensus among the GPs regarding which comments were the relevant ones. The GPs were asked to state why a given comment was not relevant. Over 90 percent of their reasons fell into the following three groups: (a) the GP disagreed with the advice, (b) the GP agreed with the principle but he would prefer to modify the recommendation to suit his practice setting or (c) the GP felt that the advice had no consequence for the decision he had to make, although he did not disagree with the underlying principle. The treating physician judged over 50 percent of the internist's comments relevant. The predominant reason that the treating physician stated believed a comment to be irrelevant or less relevant was a misunderstanding of his intentions and/or reasoning by the critiquing physician.

3.1 INTRODUCTION

A number of workers in Artificial Intelligence in Medicine have argued that, for some medical domains, critiquing the decisions of a physician could be an appropriate approach. In this critiquing model, the physician submits his intended decisions to the program. The program evaluates these decisions and expresses agreement or suggests alternatives [1,2].
Others have stressed the importance of integrating consultation systems with routine data management functions within a medical office or institution. When decision-support systems are integrated with data-management systems, providing decision support can be viewed as a byproduct of the data-management activities [3,4,5,6]. Attempts have been made to combine the critiquing approach with data-management systems resulting in systems which, from the viewpoint of the physician, act as automated medical records, but 'behind the scenes' the decisions of the physician are evaluated and, if necessary reasoned, alternatives are suggested [7,8].

When humans interact with their environment, they form models of themselves and of the environment which they are interacting with. Such internal models are known as mental models. These mental models provide predictive and explanatory power for understanding the interaction with the environment. Similarly, the physician forms a mental model of the patient whom he is treating. When another physician is asked to critique this treatment, he has to reconstruct the intentions and reasoning of the treating physician. The critiquing physician has to formulate a model of the treating physician's mental model (Figure 1). Such a model of a mental model is known as a conceptual model [9]. In the medical record, one encounters both data describing the patient's state (e.g., the results of laboratory tests) and data describing the mental model of the treating physician (e.g., a description of treatment goals). The creation of such a conceptual model of the treating physician lies at the heart of a critique: The recognition of the intentions (the treatment objectives) of the physician in combination with the actions undertaken to achieve these treatment objectives [1].

In our study, we wanted to evaluate whether the computer-based medical record of a primary-care information system contained enough information to allow another physician to create a conceptual model of the mental model of the treating physician, and subsequently to generate critique. The rationale was that the ability to reconstruct the reasoning of the general practitioner must be

a prerequisite for the development of a critiquing system.

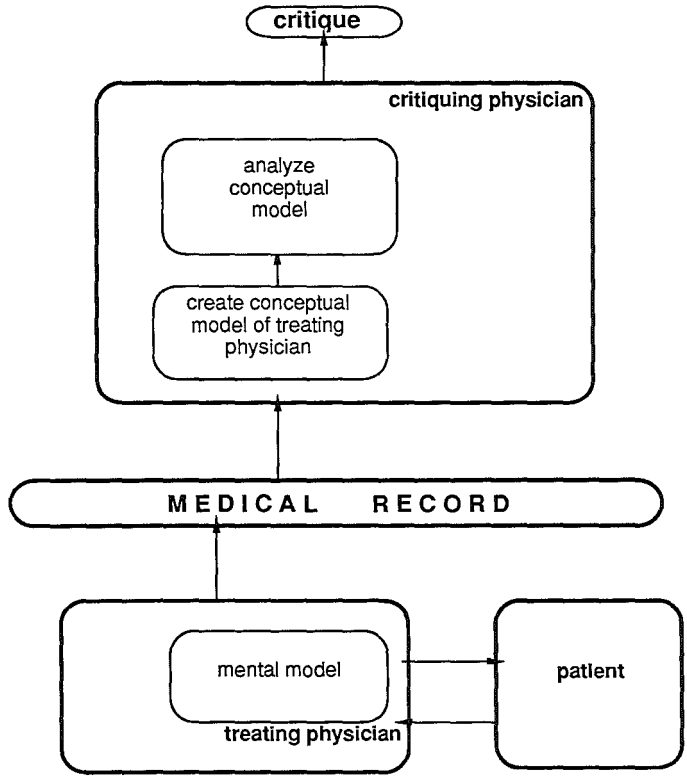

Figure 1:
The treating physician develops a mental model of the patient he treats and his role in that treatment. In the medical record data describing the condition of the patient and data describing the intentions of the treating physician are recorded. The critiquing physician has to formulate a conceptual model of the intentions of the treating physician.

3.2 MATERIAL AND METHODS

A major restriction that we imposed on the critiquing process was that the critiquing physician have only the medical record at his disposal. When the critiquing physician was in doubt about the contents of the medical record (e.g., the exact meaning of a given diagnosis), then the treating physician was not

available for clarification. This restriction stemmed from our desire to develop critiquing systems that rely on operational information systems as their source of data.

We selected the system ELIAS, a system for the General Practitioner (GP), as a source of medical data. ELIAS supports a fully automated medical record [10]: GPs using ELIAS no longer maintain paper records. Hypertension was selected as a medical domain because it is a common disorder treated both by physicians working in primary care and by hospital physicians. We had access to an internist with a recognized interest in the treatment of hypertension in primary care. One of the GP's was asked to provide us with the computer-based medical records of five patients with hypertension: patients he considered to be average patients, neither the easy cases, nor the most complex cases. The ELIAS system was introduced in the practice of this GP in the spring of 1985; the selection of patients took place in the spring of 1987. We thus had access to the computer-based medical record of the previous two years. The number of visits to the GP during these two years ranged from four to 40. Although the blood pressure was recorded at almost all visits, not all visits were related to hypertension: one patient was also treated for infections of the respiratory tract, another suffered an exacerbation of a depression, and a third developed a myocardial infarction.

The study itself consisted of three stages: During the *first stage* a printout of the computer-based medical record was submitted to the internist. The internist was not associated with the GP who made the medical records available. In 'thinking aloud' sessions, the internist was asked to review the treatment of the hypertension as documented in the medical record. His free-flowing critiques were subsequently divided into discrete 'comments'. Each comment was considered to be an individual remark -- an entity that was directed to a specific action described in the medical record (or the absence of an action). If the internist would state: "I would not treat this patient with medication A but with B, but if you insist on treating with A, then the dosage is too high", this remark would be considered as two separate comments. The comments recorded by the investigator were later handed back to the internist for review and approval.

The *second stage* of the study involved submitting the comments of the internist to three GPs who acted as a panel judging the relevance of these comments. The GPs were aware of the fact that these critiques were generated by an internist who worked in a large clinic. The GPs were not associated with the

internist or with the treating GP. Each of the three GPs was asked individually to rate each comment of the internist as either 'relevant', 'irrelevant' or 'partially relevant'. If the GP did not consider a comment relevant, he was asked to state why.

The *third stage* of the study involved sharing the critique of the internist with the same GP who made the medical records available. The GP was also asked to rate each individual comment of the internist as either 'relevant', 'irrelevant' or 'partially relevant'. If the GP did not consider a comment to be relevant, he was asked to state why.

3.3 RESULTS

The internist generated in total 48 comments. The comments ranged from recommendations for minor adjustments in the dosages of given drugs to suggestions for major revisions of the therapy plan. The investigators assigned each comment to one of three groups. The first group involved comments dealing with the detection of the cause of the hypertension and assessing the severity of the hypertension: We shall call these *diagnostic comments*. The second group involved comments dealing with the selection of the optimal treatment for the patient: We shall call these *selection comments*. The third group involved comments dealing with the execution of the treatment, the dosage to give, the precautions which should be taken, the side effect to monitor: We shall call these *execution comments*. Of the 48 comments, 18 were diagnostic comments, 13 were selection comments, and 17 were execution comments.

In the second stage of the study these comments were submitted to the three GPs individually. The results are shown in Table 1: GPa judged 15 comments as either irrelevant or only partially relevant, GPb judged 21 comments as either irrelevant or only partially relevant,and GPc judged 23 comments as either irrelevant or only partially relevant. One needs an impression of the inter-observer variability: Was there a consensus among the GPs as to what comments were the relevant ones? When a comment received the verdict 'relevant' from two out of the three GPs and the third GP judged the comment 'relevant' or 'partially relevant', we labeled that comment *accepted*. When two out of three judged the comment 'irrelevant' and the third judged it as 'irrelevant' or 'partially relevant', we labeled the comment *rejected*.

Table 1: The judgement of the panel of GPs concerning the relevance of the internist's comments.

	GPa	GPb	GPc
comment is relevant	33	27	25
comment is partially relevant	4	12	4
comment is irrelevant	11	9	19

All other comments (e.g. one GP judged the comment 'relevant' whereas the other two judged it 'irrelevant') were labeled *scattered*. The results are shown in Table 2: 18 comments of the internist fell in the category 'accepted', 26 in the category 'scattered', and four in the category 'rejected'. Of the 17 execution comments 12 were accepted, whereas of the 18 diagnostic comments only three were accepted.

Table 2: GPs judgement on internist's comments related to diagnosis, selection of treatment, execution of treatment.

	accepted	scattered	rejected
diagnostic comments	3 (17%)	14	1
selection comments	3 (23%)	7	3
execution comments	12 (71%)	5	0

When the GPs judged a comment to be irrelevant or only partially relevant, they justified their disagreement in several ways:

- The GP *disagreed* with the medical reasoning as presented by the internist.
- The GP stated that the comment had *no consequences*; the GP did not disagree with the underlying principle, but he felt that the comment was irrelevant to the decision he had to make.
- The GP agreed with the internist in principle, but the GP wanted to *modify* the recommendation to suit his own practice setting.
- The GP stated that the comment was based on an incomplete

understanding or *misunderstanding* of the intentions and/or reasoning of the treating physician.
- The reasoning could not be assigned to any of the previous groups for *miscellaneous* reasons.

The most frequent reasons for not judging a comment to be relevant were (a) agreeing with the underlying principle yet applying it somewhat differently, (b) disagreeing with the medical reasoning as presented by the internist and (c) stating that the comment of the internist had no consequences for the treatment (Table 3).

Table 3: The reasons the GPs in the panel stated for judging a comment of the internist irrelevant or only partially relevant.

	DA*	NC	BA	MU	MIS
diagnostic comments	7	9	11	2	1
selection comments	11	7	1	0	2
execution comments	0	0	7	0	1

*DA : disagreeing with the medical reasoning,
NC : the comment has no consequences,
BA : basically agreeing with the principles stated in the comment, but wanting to modify the recommendation,
MU : the internist misunderstands the treating physician,
MIS : miscellaneous reasons.

In the third stage of the study the GP who treated the patients was asked to judge the comments of the internist: He judged 25 comments relevant, 11 comments partially relevant, and 12 comments irrelevant. In seven cases, the GP disagreed with the medical reasoning, in three cases the GP agreed with the principle but wanted to modify the recommendation to suit his own practice setting, and in two cases the GP felt that the comment had no consequences for the treatment under consideration. But the predominant reason (on 10 occasions) that the treating physician stated believed a comment to be irrelevant or less relevant was a misunderstanding or incomplete understanding of his intentions and/or reasoning by the critiquing internist.

3.4 DISCUSSION

In the domain of hypertension, the printout of the computer-based medical record seems to contain enough information for a human observer to generate considerable advice. The ability to generate critique indicates only that some conceptual model can be formulated by the critiquing physician. The ability to generate critique does not prove the validity of the underlying conceptual model: The predominant reason that the treating physician believed a comment to be irrelevant was a misunderstanding of his intentions by the critiquing physician.

For example, the internist commented on the use of diagnostic tests that he thought were superfluous. The treating GP responded that the patient had requested these tests, as the patient had had a relative with hypertension who recently died of a rare disease. The patient was anxious that she might have the same ailment, and the presence of this disease could readily be excluded by blood analysis. The GP was performing the test in the context of the anxiety of the patient, whereas the internist assumed that the test was ordered in the context of a diagnostic work-up for hypertension.

Another patient developed an agitated depression. The internist pointed out that this type of patient often develops high blood pressure. Therefore, the internist recommended treatment of the depression, and, if the blood pressure would not return to normal levels, then anti-hypertensive medication might be prescribed. The GP replied that he had tried discussing this topic with the patient, but that the patient had refused psychiatric help.

In another case the internist stated that the initial dose of a drug was too low. The GP responded that the indication for prescribing this drug had not been hypertension but rather something else.

One might argue, that the working situation of the GP and the methods which the GP uses to manage this situation differ from those of the clinician [11,12,13,14]. Consequently, the internist is limited in his ability to deduce the intentions of the GP. If the difference between a GP and an internist would account for the inability of the internist to understand the GP, then one would expect that other GPs, just like the treating GP, would be able to identify these misunderstandings. Yet, the panel of three GP's failed to detect a

misunderstanding of the intentions of the treating physician by the internist as a cause of the generation of irrelevant critique.

The medical record of the GP is primarily a record of what the actions he/she performed; it is a "what did I do" record, not a "why did I do it" record. A particular prescription is not labeled "for the treatment of hypertension". Diagnostic tests are not labeled "to exclude disease X". The underlying reasoning has to be reconstructed. (Similar observations prompted Weed [15] to the development of his Problem Oriented Medical Record.) Moreover, not all actions or decisions of the GP are mentioned in the medical record. Missing data may lead to an incorrect interpretation of other data and subsequently lead to a conceptual model that, when used to generate a critique, produces 'irrelevant' advice. The word 'irrelevant' used in this context denotes a situation where the treating physician identifies the critiquing physician's incorrect assumptions, recognizes the consequences of these incorrect assumptions for the conceptual model of the internist, and subsequently disregards the advice.

A conceptual model that captures the intentions of the treating physician does not guarantee that a critique generated on the basis of that conceptual model will be judged relevant by other physicians. Clinical practice is not solely based on scientific facts and evidence. Both training and practice setting have an effect on medical decision-making [16,17,18,19]. When a physician receives a critique, he will not only verify the conceptual model underlying that critique, but he will also evaluate whether, in his practice setting and with his training, the critique is relevant; he will compare his mental model with that of the critiquing physician. The three main reasons the GPs gave for judging a comment less than relevant were: (a) the GP disagreed with the advice, (b) the GP agreed with the principle but he would prefer to modify the recommendation to suit his practice setting or (c) the GP felt that the advice had no consequence for the decision he had to make. We will consider these three reasons in turn (Figure 2).

- When the GP judges the advice of the internist to be less relevant or irrelevant because he *disagrees with the medical reasoning*, he is indicating that his mental model differs from the mental model of the internist. The GP recognizes the inferences of the internist and disagrees with those inferences. The resulting advice is subsequently judged irrelevant. The GP does not disagree with the conceptual model that the internist constructed of the treating physician's behaviour based on data in the medical record;

if the GP disagrees with the conceptual model of the internist, he would argue that the internist misunderstands the intentions of the treating physician.

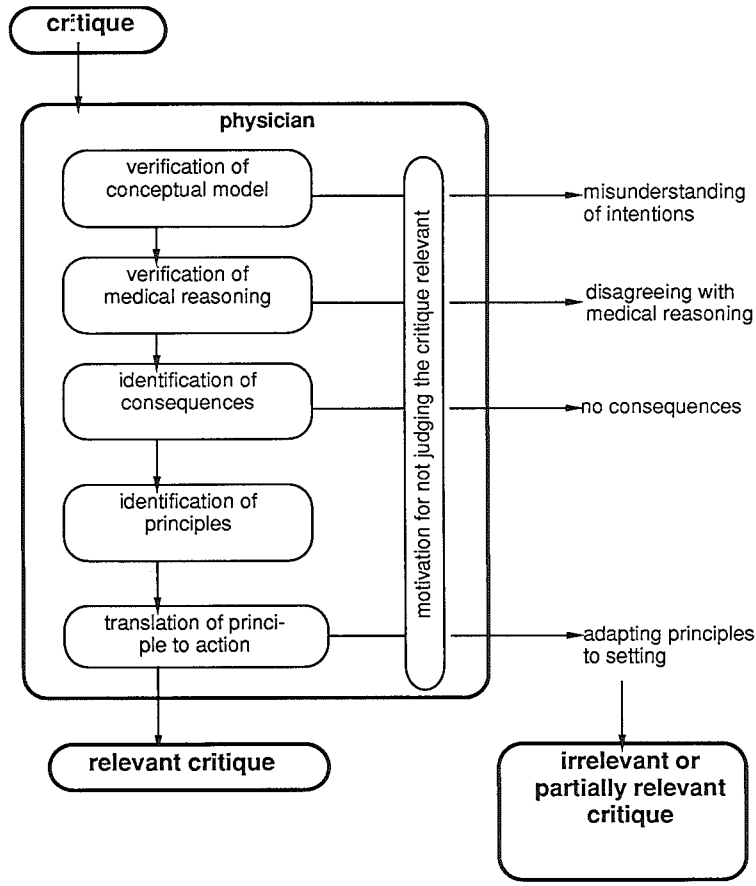

Figure 2:
The physician who receives the critique may have a variety of reasons for judging the critique irrelevant or only partially relevant. For an analysis of these reasons see text.

An example, in the case of a 48 year old male with blood pressures as high as 190/100 mm Hg, the internist argued that because of the age of the patient and the severity of the hypertension renal-artery stenosis should be considered. The GP responded that, although he agreed with the observation that the patient was young and was suffering from a severe hypertension, he disagreed with the need to investigate the possibility of

a renal-artery stenosis. He would only consider this investigation necessary if the patient would not respond to therapy. Since the patient was responding to therapy a stenosis was, in his opinion, highly unlikely.

- Another situation arises when the GP states that the advice of the internist has no *consequences* for the treatment of the patient. The GP does not disagree with the inferences made by the internist, but states that the comment has no effect on his decision making. In the majority of these comments, the internist was recommending the collection of additional data (e.g., laboratory tests) without stating how, once obtained, the data would influence the treatment. The response of the GP was, typically, that no matter what the outcome of these test, he would treat the patient in the same manner.

- When reviewing the advice of the internist, the GP may be able to distinguish treatment principles from the translation of these principles into actions. The GP may agree with the fact that the principles should be applied but may *disagree with the actions* that the internist recommends. The mental model of the GP differs from the mental model of the internist when viewed at the level of specific actions to be taken, yet the models are congruent at the more abstract level of treatment principles.

When comparing the judgments of the three GPs, one observes a *lack of consensus* on whether the advice is relevant. The comments dealing with the execution of the anti-hypertensive treatment were better received by the majority of the GPs than the comments dealing with diagnosis or the selection of treatment. The predominant reason for judging the comments dealing with the execution of the therapy as irrelevant or only partially relevant did not involve the treatment principles stated in the comments, but involved the translation of those principles to actions; the GP wanted to modify the recommendations of the internist to suit his practice setting. In the areas of diagnosis and the selection of treatment, comments are seldom rejected by all GPs, but the GPs judge the relevance of the comments differently.

3.5 CONCLUSIONS

The automated medical records did contain enough information for a human observer to generate substantial critiques. Both the treating physician and the

panel of GPs judged more than half of the critiques as relevant.

Generating a critique will often involve discussing the actions of a physician in the context of a conceptual model of that physician. But the need to create such a conceptual model of the treating physicians was not anticipated in the design or use of the automated medical record. Subsequently, the ability to generate these conceptual models is limited. Errors in an observer's conceptual model will often invalidate his critiques.

In our study, critiques dealing with the execution of a treatment are judged more frequently relevant than those dealing with a diagnosis or the selection of a treatment. Critiques dealing with the execution of treatment are often adapted by the GP to suit his setting. In this process of modifying the recommendation, the GP separates the treatment principle from the translation of that principle into action. Multiple translations of the same treatment principle into different actions are possible.

Apart from critiques dealing with the execution of a treatment, there is little consensus among the GPs on what constitutes a relevant critique.

3.5.1 Issues for the Development of Computer-Based Monitoring in Primary Care.

The wish to develop systems which produce critique as a byproduct of medical data-management activities adds several major research issues to the already non-trivial task of developing a stand-alone critiquing system.

These systems will have to reason about the intentions of the treating physician. The construction of such a conceptual model of the treating physician based on data from automated medical records may give rise to incorrect interpretations. If treatment protocols are available, then critiquing systems can be developed by monitoring, based on the automated medical record, adherence to that protocol. In settings where well-defined treatment protocols are available (e.g. in oncology), the feasibility of this approach has been demonstrated [8]. This approach hinges on the assumption it is the physician's intention to follow the protocol. The system does not need to reason about the intentions of the physician -- the protocol is used as a substitute. And, ultimately, the only critique of the system is that the physician does not follow

the specified protocol.

Moreover, the lack of consensus on what constitutes a relevant critique poses significant problems. Physicians do not request critique but receive it whenever monitored patient data warrant it. Therefore one has to avoid generating excessive amount of critique which is subsequent judged irrelevant. Too much "irrelevant critique" may not only create antagonistic responses, but may also blunt the usefulness of those critiques that have greater clinical significance. Additional research is required to establish a better understanding as to what type of critique is likely to be judged relevant.

ACKNOWLEDGMENTS

The support of the Nederlandse Hartstichting (Netherlands Heart Foundation, grant no. 88.236) is gratefully acknowledged.

REFERENCES

[1] Miller PL. Goal-directed critiquing by computer: ventilator management. *Comp Biomed Res* 1985;18:422-38.

[2] Miller PL. *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer.* New York: Springer Verlag, 1986.

[3] Shortliffe EH. Computer programs to support clinical decision making. *JAMA* 1987;258:61-6.

[4] McDonald CJ, Hui SL, Smith DM, et al. Reminders to physicians from an introspective computer medical record. *Ann Intern Med* 1984;100:130-8.

[5] Warner HR. *Computer-Assisted Medical Decision Making.* New York: Academic Press Inc, 1978.

[6] Pryor RA, Gardner RM, Clayton PD, et al. The HELP system. *J Med Syst* 1983;7:87-102.

[7] Evans RS, Larsen RA, Burke JP, et al. Computer surveillance of hospital acquired infections and antibiotics. *JAMA* 1986;256:1007-11.

[8] Langlotz CP, Shortliffe EH Adapting a consultation system to critique user plans. *Int J Man-Machine Stud* 1983;19:479-96.

[9] Gentner D, Stevens AL, eds. *Mental Models.* London: Lawrence Erlbaum Associates, 1983.

[10] Westerhof HP, Boon WM, Cromme PVM, et al. ELIAS: Support of the Dutch General Practitioner. In: Reichertz PL, Engelbrecht R, Picollo U, eds. *Present Status of Computer Support in Ambulatory Care.* New York: Springer Verlag, 1987, 1-10.

[11] Pendleton D, Hasler J. *Doctor-Patient Communication.* London: Academic Press, 1983.

[12] Bentsen BG. Fundamentals of general practice. *Scand J Primary Health Care* 1984;2:11-7.

[13] Perlman LV, Graham T, Christy W. Primary care internal medicine residencies: definitions, problems and opportunities. *Arch Int Med* 1976;136:111-113.

[14] Fry J. ed. *Primary Care.* London: Heinemann, 1980.

[15] Weed L. *Medical Records, Medical Education and Patient Care.* Cleveland: Case Western Reserve University Press, 1971.

[16] Palchik NS, Dielman TE, Woolloscroft JO, et al. Practice preferences of primary care and traditional internal medicine house officers. *Med Educ* 1987;21:441-9.

[17] Petersdorf RG. The doctor's dilemma. *N Eng J Med* 1978;299:628-34.

[18] Goldenberg DL, Pozen JL, Cohen AS. The effect of primary-care pathway on internal medicine residents' career plans. *Ann Intern Med* 1979;91:271-4.

[19] Weil PA, Schleiter MK. National study of internal medicine manpower: VI. Factors predicting preferences of residents for careers in primary care or subspeciality care and clinical practice or academic medicine. *Ann Intern Med* 1981;94:691-703.

CHAPTER 4

A Model for Critiquing Based on Automated Medical Records

Submitted for publication

Johan van der Lei
Mark A. Musen

ABSTRACT

We describe the design of a critiquing system, HyperCritic, that relies on automated medical records for its input data. The purpose of the system is to advise general practitioners who are treating patients who have hypertension. HyperCritic has access to the data stored in a primary-care information system which supports a fully automated medical record. HyperCritic relies on data in the automated medical record to critique the management of hypertensive patients, avoiding a consultation-style interaction with the user.

The first step in the critiquing process involves the interpretation of the medical record in an attempt to discover the physician's actions and decisions. After detecting the relevant events in the medical record, HyperCritic views the task of critiquing as the assignment of critiquing statements to these patient-specific events. Critiquing statements are defined as recommendations involving one or more suggestions for possible modifications in the actions of the physician. The core of the model underlying HyperCritic is that the process of generating the critiquing statements is viewed as the application of a limited set of abstract critiquing tasks. We distinguish four categories of critiquing tasks: preparation tasks, selection tasks, monitoring tasks, and responding tasks. The execution of these critiquing tasks requires specific medical factual knowledge. This factual knowledge is separated from the critiquing tasks and is stored in a medical fact base.

The principle advantage demonstrated by HyperCritic is the adaption of a domain-independent critiquing structure. We show how this domain-independent critiquing structure can be used to facilitate knowledge acquisition and maintenance of the system.

4.1 INTRODUCTION

Many medical decision-support systems rely on a consultation model for their interaction with the user. In the consultation model, the program serves as an adviser, accepting patient-specific data, asking questions, and generating advice for the user about diagnosis or therapeutic management [Barnett et al., 1987; Miller and Masarie, 1990; Shortliffe, 1987]. Certain workers in medical informatics have argued that, for some medical domains, critiquing the decisions of a physician is a preferred approach in providing decision support. In this critiquing model, the physician submits to the program, in addition to patient-specific data, the decisions he intends to make. The program evaluates these decisions and expresses agreement or suggests alternatives [Miller, 1986]. Other researchers have stressed the importance of integrating consultation systems with routine data-management functions within a medical office or institution. When decision-support systems are integrated with data-management systems, provision of decision support can be viewed as a byproduct of the data-management activities [McDonald et al., 1984; Pryor et al., 1983; Warner, 1978]. Other workers have attempted to combine the critiquing approach with data-management systems, resulting in systems that, from the viewpoint of the physician, act as automated medical records, but that "behind the scenes" evaluate the decisions of the physician and, if necessary, suggest reasoned alternatives [Evans et al., 1986; Langlotz and Shortliffe, 1983].

This paper describes the design and implementation of a critiquing system that relies on automated medical records for its data input. The purpose of the system is to offer comments to general practitioners on the treatment of hypertension. The system, HyperCritic, has access to the data stored in a primary-care information system that supports a fully automated medical record. A major restriction that we have imposed on HyperCritic is that the program must rely solely on the automated medical record for data input; a consultation-style interaction with the user is avoided.

Underlying any computational system is a model of a particular domain. Models are, by their nature, selective in the entities that they contain, as all models are developed from a particular perspective. In this paper, we shall explain the importance of abstracting to an appropriate level the domain in which the system functions. We shall discuss our perspective regarding the process of

critiquing hypertension therapy, and describe the critiquing model that we developed. To validate our ideas, we created HyperCritic. We shall provide details concerning the computational implementation of HyperCritic, and shall examine examples of the system's output. A clinical evaluation of HyperCritic has been presented elsewhere [Van Der Lei and Musen, 1990]. In the description of the model and of its implementation, we shall contrast our approach with those taken by developers of other systems that provide decision support based on automated medical records.

An architecture for a medical decision-support system that concentrates on modeling the application tasks to be performed generally is referred to as a *task-based architecture*. Several researchers have noted the difference between the procedural aspects of a task and the specific knowledge required to execute that task [Fox, 1989; Gruber, 1988; Hasling et al., 1984; Lanzola et al., 1989; Musen, 1989b; Swartout, 1981;]. The unifying theme in these research projects is the notion that the procedural aspect of a given task should be represented separately from the specific knowledge required to execute that task. Different researchers, however, have addressed the issue of separating these knowledge components from different perspectives. Consequently, the knowledge components that they have identified differ, and the systems that they have developed illustrate different advantages that can be gained from separating these knowledge components.

The representation of the problem-solving behavior of NEOMYCIN on the level of tasks greatly enhanced the explanation facilities of that system because it provided explanations in terms of the tasks that need to be performed in that domain [Clancey and Letsinger, 1981; Hasling et al., 1984]. Musen [1989b] showed in the PROTÉGÉ system how the separate modeling of a task's process and content components can be used to develop knowledge-acquisition tools. The developers of the Oxford System of Medicine showed how the same medical knowledge can be used for a variety of tasks [Glowinski et al., 1989; Fox, 1989].

The work presented in this paper describes the application of task-based architectures to the problem of critiquing based on automated medical records. In the domain of critiquing systems, the advantages of task-based architectures have not been explored. We propose a task-based model for critiquing physicians' management of patients based on data from automated medical records. In the HyperCritic program, we explore the advantages of the resulting

task-based architecture for a critiquing system in the domain of the therapeutic management of patients who are hypertensive.

4.2 GENERATION OF COMPUTATIONAL MODELS

Underlying any computational system is a model of the task domain of that system and of the methods by which problems in that domain are addressed. For example, a hierarchical database presumes a data model based on hierarchical relationships among data elements, whereas a relational database permits system builders to describe a variety of tabular relationships among data elements. The form of the model subsequently dictates the terms and relationships used in the system. In the case of the relational database, the relational model is expressed in terms of tables in the database and of the operations (for example, select, project, and join) that manipulate the data stored in these tables. In contrast to the explicit data models that undergird database systems, the domain models underlying decision-support systems often are implicit. The behavior of such systems is described in terms of the symbols (for example, rules, frames, or objects) and the inference strategies that manipulate those symbols (for example, forward chaining, backward chaining, or belief update in causal probabilistic networks) [Musen and Van Der Lei, 1989].

An important perspective articulated by Newell [1982] is that knowledge is an abstraction that can be separated from the symbols that are used to represent the knowledge. Knowledge is a set of goals and the behavior potentially needed to achieve those goals. Knowledge itself can never be written down; it can only be observed as an activity. This distinction between knowledge (at what Newell refers to as the *knowledge level*) and the symbols used to represent knowledge (at the *symbol level*) allows us to distinguish our goals for an intelligent system from the language that we use at the symbol level to represent these goals. Thus, knowledge-level analysis of an application task specifies the behaviors that are required to solve a problem in the world; analysis of a knowledge base at the symbol level specifies the computational mechanisms needed to model the requisite behavior. Researchers in artificial intelligence (AI) increasingly agree that it is important to understand a domain task in terms of its knowledge-level specifications before proceeding to a symbol-level implementation. Although

there is little consensus about how to go about describing domain tasks at the knowledge level, the goal becomes to understand a system's behavior in terms of an abstract model, rather than by means of a specific set of notations [Clancey, 1985; Newell, 1982].

4.3 A MODEL FOR CRITIQUING BASED ON AUTOMATED MEDICAL RECORDS

In this section, we shall present a model for critiquing physicians' management of hypertension based on automated medical-record data. To clarify the concepts used, we shall illustrate how similar notions are embedded in other decision-support systems that rely on automated medical records for data input. To avoid confronting the reader with the variety of notations used in different programs, we shall draw all our illustrations from the well-known CARE system [McDonald, 1981].

The CARE system provides a syntax for writing rules that remind the physician of diagnoses or problems that she might otherwise have overlooked. Typically, these rules use simple logic, and cause fixed paragraphs to be displayed as a standard response to a definite or potential abnormality. Although we shall describe the purposes of a number of rules from CARE in terms of abstract critiquing tasks, CARE itself has no explicit representation of these abstractions.

4.3.1 Critiquing Knowledge Versus Medical Knowledge

The foundation for the critiquing approach hinges on the observation that medical practice is characterized both by practice variation and by subjective evaluation [Miller, 1986]. Therefore, an objective standard for the ideal treatment often is not available. If such an objective standard is available, then a system can be developed that embodies that standard (for example, in the case of oncology protocols [Tu et al., 1989]). In the absence of such an objective standard, the critiquing approach advocates a discussion of the relative merits of a proposed treatment plan. The physician is asked to type into the computer her treatment plan together with a patient description. The system's response consists of comments pointing out conflicts between the condition of the patient and the proposed treatment plan; the program may also suggest alternatives.

The core of our model is that two distinct types of knowledge are needed during the critiquing process: knowledge about the process of critiquing itself (that is, *critiquing knowledge*) and specific medical knowledge that will be required during the critiquing process (that is, *medical knowledge*). Critiquing knowledge describes when and how to critique, whereas medical knowledge provides the factual knowledge needed during the critiquing process. Critiquing knowledge describes the process of critiquing; medical knowledge describes the content of critiquing.

```
IF NO "BETA BLOCKERS USE"
   THEN EXIT
:
   IF      "AV BLOCK D" WAS ON AFTER "MOST RECENT VISIT"
      OR   "D LAST" WAS = "RAYNAUDS PHENOM"
   THEN    Review use of "Beta Blockers". Contraindication exists. R:1457.
:
```

Figure 1: Rule from CARE that Reports a Contraindication.
A decision rule from CARE [*Source:* McDonald, 1981, page 84, used with permission] that warns the physician that the presence of Raynaud's disease is a contraindication for the use of beta blockers.

Several rules in CARE deal with warnings related to contraindications for specific drugs. Figure 1 shows a statement from CARE that warns the physician that the presence of Raynaud's disease is a contraindication for the use of beta blockers. The general form of these rules is:

If drug A is started;
and finding B is present;
then report that finding B is a contraindication for drug A.

We can remove the specific medical knowledge from this rule:

For any drug that is started;
get the known contraindications of that drug;
  For all known contraindication of that drug;
  determine whether that contraindication is present;
  If that contraindication is present;
  report the contraindication.

Observe that, although we have removed specific medical knowledge, this procedure assumes the presence of medical knowledge describing contraindications. To execute the procedure, we need a database from which to obtain potential contraindications to the therapy. The representation of the contraindications within that database must be such that the computer can subsequently verify the presence of those contraindications in the medical record. Note that the procedure does not assume the presence of a specific contraindication, but rather assumes a uniform representation of contraindications, together with a mechanism for retrieving those contraindications that meet a given criterion (in our example, *get the known contraindications of that drug*). When we describe critiquing knowledge (the knowledge describing the process of critiquing), we are forced to specify the medical knowledge that will be required during the critiquing process in terms of abstract notions that represent classes or types of medical knowledge.

Observe also that the notion *any drug that is started* is a generalization of *drug A is started*. In order to remove the specific medical knowledge from the initial rule from CARE (see Figure 1), the specific action of the physician (starting a beta blocker) has to be replaced with an abstract notion identifying a class of possible actions of the physician (starting any one of various drugs).

These three concepts -- (1) a class of physician actions representing the notion of starting a drug, (2) contraindications, and (3) a mechanism to establish whether contraindications apply in special situations -- create a structure that enables the definition of a procedure that is generalized over all the possible actions within that class of actions.

We shall first elaborate on the notion of a class of physician actions; then we shall further develop the notions of critiquing knowledge and of medical knowledge.

4.3.2 The Actions and Decisions of the Physician

If manual entry of information into a critiquing system is to be replaced with automated entry of data from an electronic medical record, then, in the absence of direct interaction between physician and system, the critiquing system will have to interpret the computer-based medical record so as to discover the treatment plan. McDonald points out an important limitation of medical records

when he states that the computer-based medical record is "only a pale reflection of the patient's true state" [Source: McDonald, 1981, page 11]. Moreover, the medical record is also only a pale reflection of the state of the physician's decision making [Musen, 1989a; Van der Lei et al., 1989]. None the less, the first step in critiquing is to interpret the medical record to discover the actions or decisions that constitute the physicians' treatment (see Figure 2).

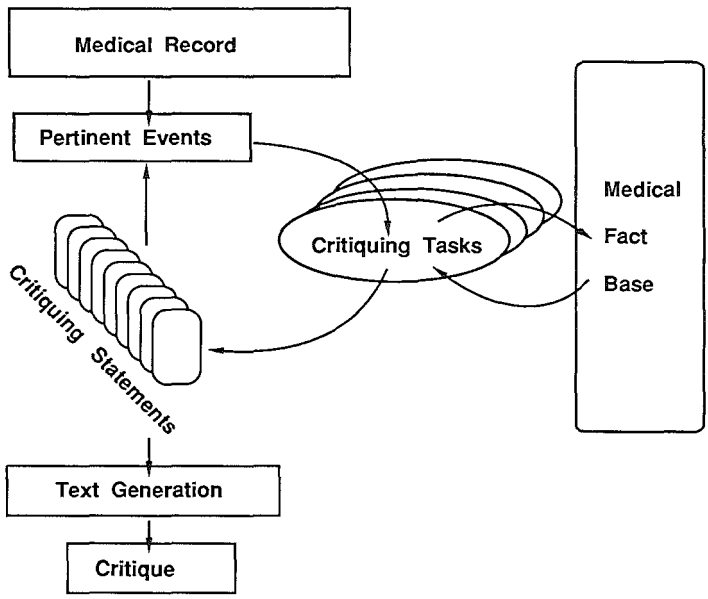

Figure 2: The Critiquing Process
The first step the critiquing process involves the interpretation of the medical record in an attempt to discover the physician's actions and decisions: the pertinent events. The critiquing process involves associating critiquing statements with these pertinent events. The core of the model is that the process of generating the critiquing statements requires two types of knowledge: critiquing knowledge and medical knowledge. Critiquing knowledge is represented as a limited set of critiquing tasks. The execution of those critiquing tasks requires specific medical knowledge. This medical knowledge is separated from the critiquing tasks and is stored in a medical fact base.

To describe critiquing knowledge (the knowledge that specifies the process of critiquing), we need to introduce abstract notions that identify whole classes of possible physician actions. We can, subsequently, specify critiquing knowledge that is generalized over all the actions within a specific class. One such class

of actions involves "starting a drug," and the critiquing knowledge that applies to all possible actions within that class (that is, to actions of starting specific drugs) can indicate that a search for contraindications must be performed. We refer to these abstract notions that describe classes of actions as *event descriptions*. When a physician's action falls within a given class of actions identified by an event description, we say that the physician's action *matches* that event description.

We can thus search in the medical record for the physician actions that match event descriptions. When we find such a physician action, we refer to it as the *pertinent event* that matches the event description. For example, treating hypertension often involves prescribing drugs. Event descriptions in the hypertension domain include, among others, "starting a drug," "increasing the dose of a drug," and "adding one drug to another." Based on these event descriptions, pertinent events can be detected: the physician starts propranolol, increases the dose of captopril, or adds propranolol to hydrochlorothiazide.

In summary, the event descriptions are abstract notions that identify classes of possible actions or decisions of a physician in a given domain. The pertinent events represent the physician's actual actions or decisions at a specific moment. When we define critiquing knowledge, we use the event descriptions to specify the classes of actions for which that critiquing knowledge is valid.

4.3.3 Critiquing Knowledge

After the pertinent events in the medical record have been detected, the task of critiquing is viewed as the assignment of critiquing statements to those pertinent events. *Critiquing statements* are recommendations involving one or more suggestions for possible modifications in the actions of the physician. Another key notion in our model is that critiquing knowledge can be viewed as the application of a *limited set of critiquing tasks*. Each such critiquing task (1) identifies a class of physician' actions, (2) identifies a potentially appropriate critiquing statement, and (3) provides a procedure that should be executed to determine whether the specified critiquing statement is valid.
In the example in Section 4.3.1, we removed factual medical knowledge from rules that dealt with contraindications. We can specify a critiquing task that represents the critiquing knowledge as follows:

applies to:	any drug that is started
critiquing statement:	existence of a contraindication
procedure:	Get the known contraindications of that drug; for all known contraindications of that drug; determine whether that contraindication is present.

We refer to the representation of such a critiquing task as the *task specification*. We refer to the total set of task specifications as the *task structure*. Note that the task specifications pose requirements that the system builder has to meet when he develops a structure representing the medical knowledge. The task specifications assume the existence of certain classes of concepts representing medical entities (for example, contraindications), and the mechanisms that operate on those concepts. We refer to this medical knowledge as the *medical fact base*.

4.3.4 Critiquing Tasks

A key notion in our model is that critiquing knowledge can be viewed as the application of a limited set of critiquing tasks. We distinguish four types of critiquing tasks: *preparation tasks*, *selection tasks*, *monitoring tasks*, and *responding tasks* (Table 1). In the following sections, we shall discuss each type of task individually, and shall describe rules from CARE in terms of it. We provide these examples to illustrate that rules from CARE can be viewed as one possible implementation of the more general notion of critiquing tasks. It should be remembered, however, that CARE does not have an explicit representation of these abstractions. The principle advantage demonstrated by the HyperCritic program is the adaption of a *domain-independent critiquing structure*.

4.3.4.1 Preparation Tasks

The use of preparation tasks hinges on the notion that certain pertinent events (actions or decisions of the physician) should be preceded by one or more preparations or observations. Execution of these critiquing tasks begins by

Table 1: Critiquing Tasks in HyperCritic

Type of Task	Contents of Medical Fact Base	Reports
preparation	preparations associated with physician's actions	absence of preparation
selection	selection criteria associated with physician's actions	presence of selection criteria
monitoring	monitoring requirements associated with physician's actions	absence of monitoring
responding	response-requiring situations associated with physician's actions	presence of response-requiring situation

tracing the preparations associated with a particular event in the medical fact base. Subsequently, the program that executes these tasks verifies in the medical record whether the required preparations indeed did take place. If they did not take place, then their absence is reported to the physician as a critique (see Table 1).

```
IF    "THYROID UPTAKE LAST" WAS GT 45
      AND ON_AFTER "MOST RECENT VISIT"
THEN
   IF       "FEMALE"
      AND   AGE IS LT 50
      AND   NO "HYSTORECTOMY SURG"
   THEN     Be certain patient is not pregnant & advise to use absolute
            contraception if "RAI 131 rx" is anticipated (692)
   AND EXIT
```

Figure 3: Rule from CARE that Reports a Preparation.
A decision rule from CARE [*Source:* McDonald, 1981, page 309, used with permission] that warns the physician that therapy with radioactive iodine, which is potentially teratogenic, should always be accompanied by the use of absolute contraception.

The CARE rule in Figure 3 could alternatively be represented in terms of a preparation task. The action of the physician (therapy with radioactive iodine, which is potentially teratogenic) is associated with a preparation (the use of absolute contraception).

4.3.4.2 Selection Tasks

Selection tasks embody the notion that there are situations in which certain pertinent events may not be the most appropriate actions or decisions. Execution of these tasks begins by tracing the situations in which the pertinent event is not the most appropriate. Subsequently, the program that executes these tasks verifies in the medical record if these situations are present. If one of them is present, it is reported to the physician in a critique.

The example from CARE shown in Figure 1 could be described in terms of a selection task. The pertinent event (prescribing a beta blocker) is associated with a situation that indicates that this pertinent event is less appropriate (the existence of Raynaud's disease). That situation, if present, will be reported.

4.3.4.3 Monitoring Tasks

Monitoring tasks represent the notion that a given pertinent event may require certain actions (for example, observing a particular patient parameter) at particular intervals. The execution of these tasks begins by tracing the monitoring requirements for a particular event, and subsequently verifying in the medical record whether these monitoring requirements are met. If they are not met, their absence is reported to the physician as a critique.

The rule in Figure 4 could be presented in terms of a monitoring task. The rule

```
IF      "PEPTIC ULCER DISEASE D"
  OR    ("INFLAMMATORY BOWEL D"
         AND NO "COLECTOMY SURG TOT")
THEN    Consider yearly screen of stool for "occult blood" to monitor GI tract
        bleeding risk.
AND  OBSERVE "OCCULT BLOOD"
AND  ORDER "OCCULT BLOOD"
  :
```

Figure 4: Rule from CARE that Reports Monitoring Requirements.
A decision rule from CARE [*Source:* McDonald 1981, pg.155, used with permission] that recommends annual screening for occult blood of patients who have a history of peptic ulcer disease or inflammatory-bowel syndrome on a yearly basis.

recommends annual screening for occult blood in patients who have a history of peptic ulcer disease or inflammatory-bowel syndrome. The absence of the recommended screening is reported to the physician.

4.3.4.4 Responding Tasks

Responding tasks incorporate the notion that some finding related to a pertinent event may require a response. The execution of these tasks begins by tracing the situations to which a response will be required. Subsequently, the program that executes the tasks verifies the presence of the situations in the medical record. If such a situation is discovered, then its presence will be reported. Note the difference between monitoring tasks and responding tasks: Monitoring tasks report the absence of a particular patient parameter and recommend observation of that parameter, whereas responding tasks report an abnormal or changed value of a parameter and recommend that the physician respond to correct the abnormal value.

The rule in Figure 5 represents an encoding for a responding task. Beta blockers may increase the frequency and severity of asthma attacks. If the program notes an increase of asthma attacks, then it reports the presence of that finding.

```
IF NO "BETA BLOCKERS USE"
   THEN EXIT
:
   IF    LAST CHANGE OF "ASTHMA SPLS" WAS ON_AFTER "MOST RECENT
         VISIT"
      AND  GT 2
      AND  AFTER "BETA BLOCKERS START"
   THEN Note "Beta Blockers" may be cause of recent increase in asthma
         attacks. R:1825 (p 1236).
:
```

Figure 5: Rule from CARE that Reports a Side Effect.
A decision rule from CARE [*Source:* McDonald, 1981, page 84, used with permission] that warns the physician that an increase in the severity and frequency of asthma attacks can be caused by beta blockers.

In summary, critiquing based on automated medical records is viewed as an interpretation of the medical record in order to detect the physician's actions and decisions, the pertinent events. This interpretation is followed by the invocation of a limited set of critiquing tasks. These tasks are designated to detect conflicts between the inferred condition of the patient and the recorded decisions the physician has made. The structure of these critiquing tasks can be separated from the actual medical knowledge required to execute those tasks.

4.4 THE HYPERCRITIC IMPLEMENTATION

To validate our ideas, we developed a system, called HyperCritic, that can critique the decision making of general practitioners caring for patients who have hypertension. Unlike previous decision-support programs that critique medical management, HyperCritic uses the notion of abstract critiquing tasks to structure the medical knowledge encoded in the system. We shall illustrate the advantages of these additional levels of abstraction in two areas: knowledge acquisition and knowledge maintenance. We first describe the current implementation of HyperCritic, and provide examples of the system's output.

4.4.1 ELIAS System

HyperCritic uses ELIAS, an automated ambulatory-medical-record facility designed for use by primary-care physicians [Westerhof et al., 1987]. ELIAS is available commercially in The Netherlands, and currently has been installed at over 200 sites. For an example of an ELIAS medical record, see Figure 6. ELIAS supports a wide range of capabilities, and physicians who use the system typically no longer maintain paper-based medical records.

A video display unit is located on the desk of the physician who enters patient data via keyboard input. Although not all data in ELIAS are coded, laboratory data and prescriptions are always stored in a machine-interpretable format. Measurements obtained during physical examination (blood pressure, pulse rate, and so on) are coded and time stamped. Prescriptions are entered under either a brand name or a generic name. A complete drug database of all

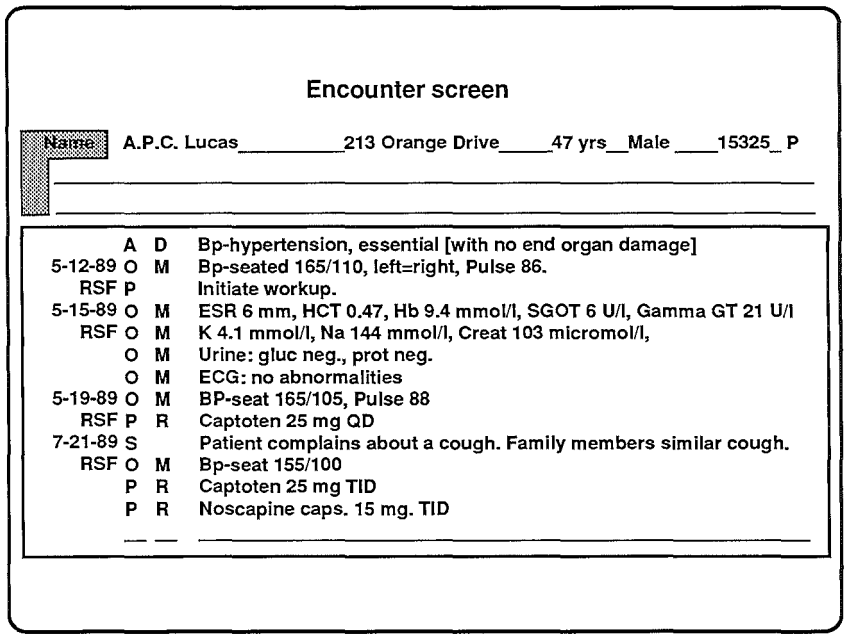

Figure 6: Encounter screen from ELIAS.
ELIAS, an information system for primary care, supports an automated medical record. Shown here is the encounter screen used by the General Practitioner to enter data during a consultation.

medications available in the Netherlands is online, and brand names can be translated to their generic equivalents. Starting and stopping dates are available for all prescriptions. One of two diagnosis coding schemes may be used at a given ELIAS site: ICPC (International Classification for Primary Care, [Lamberts, 1987]) and the coding scheme of the RCGP (Royal College of General Practitioners, [RCGP, 1984]). A problem-oriented registration [Weed, 1971] is supported, although use of the problem-oriented approach is not mandatory.

HyperCritic takes as input the electronic ELIAS medical record for a single patient. HyperCritic is activated by the presence of the diagnosis of hypertension in the medical record. HyperCritic itself does not attempt to assign the diagnosis of hypertension based on primary data in the patient's ELIAS chart.

4.4.2 Patient Description

The first stage of the process of generating a critique occurs when HyperCritic translates the primary data from the ELIAS medical record into its own patient description. The system creates the patient description by mapping the data elements of the medical-record system to a set of predefined patient parameters (vital signs, laboratory data, and so on). For several reasons, this patient description is introduced as a layer between the actual coded data in the medical record and the critiquing system. As mentioned in the previous section, ELIAS supports different coding schemes. Consequently, a given medical condition (for example, a particular symptom) may be known under different codes. Moreover, different physicians will obtain test results from different laboratories. Each laboratory will have its own normal values; consequently, a given value may be normal for one laboratory but abnormal for another. Finally, the prescriptions entered as brand names have to be mapped to their generic equivalents.

Conversion of the primary data from the ELIAS chart to the patient description involves the execution of algorithms that describe the mapping between that entity in the patient description and the observed data in the medical record. Like the data in the medical record, all diagnoses, symptoms, and quantative patient parameters in the patient description are time stamped. The value assigned to a given diagnosis or symptom in the HyperCritic patient description is one of *absent*, *unlikely*, *likely*, or *present*. A value of *absent* or *present* indicates that an unambiguous mapping between the corresponding entity in the patient description and the data in the medical record could be established (that is, based on the data in the medical record, the diagnosis or symptom could be negated or confirmed with certainty). The values of *likely* and *unlikely* are reserved for those cases in which such an unambiguous mapping was not possible (for example, when the system infers from a prescription for ibuprofen that the symptom musculoskeletal pain is likely). The value of a given quantative patient parameter in the HyperCritic patient description is one of *decreased*, *borderline decreased*, *normal*, *borderline increased,* or *increased*. The treatment history consists of generic drug names and, for each drug, the dose, the date the drug was started and, if known, the date that the drug was stopped.

4.4.3 Event Descriptions

HyperCritic must detect the pertinent events (the physician's actions and decisions) that will be critiqued at a given visit. We refer to the visit that will be critiqued as the *current visit*. To detect the pertinent events at the current visit we have created a language for the representation of event descriptions, the *event language*. HyperCritic assesses the treatment of hypertension on a visit-by-visit basis. Thus, the primitives in the event language permit the translation of prescriptions for antihypertensive drugs into clinically meaningful events based on temporal relationships among prescriptions issued at each visit (Table 2A). These primitives allow HyperCritic to compare prescriptions issued at the current visit with those issued on previous visits. Additional event descriptions are defined in terms of these primitives (Table 2B).

Table 2A: Primitives in Event Language

Primitives	**Semantics**	**Pertinent Event**
is-stopping (drug)	Has drug (or a member of the family of drugs drug) been stopped?	stopping-event
is-starting (drug)	Has drug (or a member of the family of drugs drug) been started?	starting-event
is-decreasing (drug)	Has the dose of drug (or a member of the family of drugs drug) been decreased?	decreasing-event
is-increasing (drug)	Has the dose of drug (or a member of the family of drugs drug) been increased?	increasing-event
is-continuing (drug)	Has drug (or a member of the family of drugs drug) been continued in the same dosage?	continuing-event

For example, suppose that the medical record shows that, on a certain visit, the physician stopped a prescription for the beta blocking drug metoprolol and started a prescription for pindolol, a beta blocker with properties somewhat different from those of metoprolol (Figure 7). Based on the primitives in the

Table 2B: Nonprimitives in Event Language

Nonprimitives	**Translation**	**Pertinent Event**
is-prescribing (drug)	(is starting (drug) *OR* is increasing (drug) *OR* is-decreasing (drug) *OR* is-continuing (drug))	prescribing-event
is-initializing (drug)	(is-starting (drug) *AND* (*NOT* is-stopping (drug)))	initializing-event
is-terminating (drug)	(is-stopping (drug) *AND* (*NOT* is-starting(drug)))	terminating-event
is-combining (drug1with drug2)	(is-prescribing(drug1) *AND* is-prescribing(drug2))	combining-event
is-adding (drug2 to drug1)	(is-initializing(drug2) *AND* (is-decreasing(drug1) *OR* is-continuing(drug1) *OR* is-increasing(drug1)))	adding-event
is-replacing (drug1 with drug2)	(is-stopping(drug1) *AND* is-starting(drug2))	replacing-event

event language and on combinations of these primitives, HyperCritic can identify a variety of pertinent events, such as "starting pindolol," "stopping metoprolol," "starting a beta blocker," "stopping a beta blocker," "starting a drug," "stopping a drug," "replacing metoprolol with pindolol," and "replacing a beta blocker with a beta blocker". When a pertinent event is noted, it is linked to the underlying data elements in the HyperCritic patient description or to other pertinent events. These pertinent events will subsequently serve as triggers for HyperCritic's critiquing tasks and also will serve to store the conclusions of those tasks.

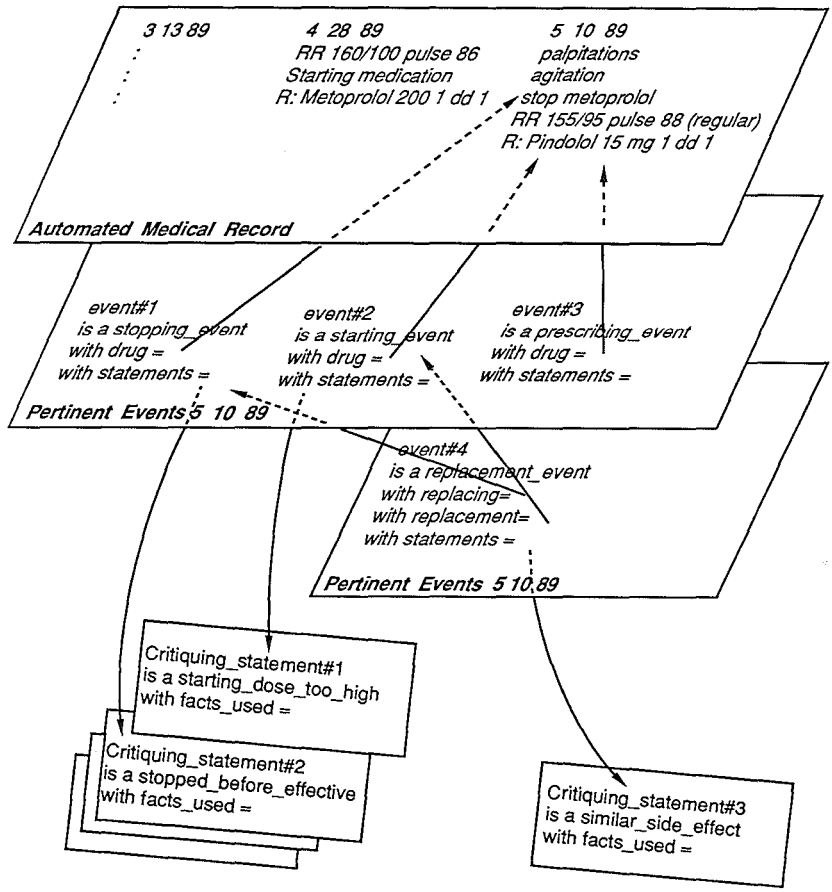

Figure 7: The Detection of Pertinent Events
HyperCritic detects the pertinent events in the medical record. These pertinent events trigger the critiquing tasks. The results of those critiquing tasks (the critiquing statements) are associated with the pertinent events.

4.4.4 Medical Fact Base

The model that we present in this paper emphasizes the distinction between medical knowledge on the one hand, and critiquing knowledge on the other hand. The critiquing tasks specify how the process of critiquing is to be performed. The critiquing tasks assume that medical knowledge is available in the medical fact base. For example, the task that screens for side effects of drugs assumes that system builders will have entered knowledge to describe those side effects. The tasks that evaluate a physician's monitoring activities assume the presence of knowledge that defines the relationships among possible adverse effects of drugs and potential monitoring activities. This medical knowledge, required by the critiquing tasks, constitutes the medical fact base. When and how this medical fact base is to be used is specified by the critiquing task structure.

All knowledge in the medical fact base is represented by frame hierarchies encoded in an object-oriented programming language. A taxonomy assigns the individual drugs used in the treatment of hypertension to specific families of drugs. Recorded for each drug are the minimum dose, the maximum dose, the increments in dose, and the time that typically is required before the therapeutic effect is seen. Special starting sequences (in which the initial dose of the drug is increased in small but rapid steps) or stopping sequences (performed to prevent the drug from being discontinued abruptly) can be specified.

The contraindications for the drugs are recorded in separate frames, indicating the degree to which the drug is contraindicated (values are *absolute*, *strong*, *relative*, *mild*). Other frames specify the possible side effects of the drugs. For each side effect, the frequency of occurrence is recorded (values are *rarely*, *sometimes*, *often*, or *very often*) together with the clinical significance (values are *very significant*, *significant*, or *temporary*). Included is an indication of whether the side effect is well established (values are *definite*, *strong evidence*, *some evidence*, or *suspected only*). Other frames specify for the side effects the monitoring activities that the physician should perform: the patient parameter that should be monitored, a time interval indicating when that parameter should be measured, and an indication of its importance (values are *mandatory*, *strongly recommended*, *recommended*, or *suggested*).

The possible causes of hypertension are not represented in the medical fact base. The purpose of HyperCritic is to monitor the treatment of hypertension,

not to search for possible causes of that condition. However, so that the system can verify that at least a minimum diagnostic workup has been performed, the requirements for a minimum hypertension workup are stored. Particular patient conditions are specified in these frames, as is the description of the required measurements or observations in case these conditions are true, together with an indication of the period for which measurement of those conditions may be considered valid. Optionally, a sentence of free text may be associated with each minimum-workup requirement that describes how the physician should respond in case the obtained measurement is abnormal. This text will be included in the output of the system.

Other frames describe for particular drugs or for families of drugs both the situations in which the drugs are not appropriate treatment, and the combinations of drugs that should be avoided. Optionally, a sentence of free text may be associated that further explains why the selection or combination is inappropriate; this sentence will be included in the output of the system.

Finally, the WHO-classification of blood-pressure levels [Gross et al., 1984] and a definition of the minimum requirements for the time between visits are available.

4.4.5 Critiquing Tasks

Critiquing knowledge in HyperCritic is represented by the abstract critiquing tasks described in Section 3.5. These critiquing tasks specify when and how HyperCritic will use the medical knowledge represented in the medical fact base to generate a critique. Such critiquing tasks require three components: (1) a triggering event, (2) a task procedure, and (3) a critiquing statement. The *triggering event* consists of an event description that identifies the type of pertinent events that will trigger the execution of the task. The *task procedure* queries the patient description and the medical fact base, and specifies the sequence of operations that is required to decide whether a critiquing statement should to be created. Executing the task procedure yields a value that is either true or false. If the result of the task procedure is true, the *critiquing statement* is created and assigned to the pertinent event that triggered the task.

For example, when the physician increases the dose of an antihypertensive drug, a critiquing task that is triggered by the event "increasing an antihypertensive drug" is processed. For each pertinent event that meets this event description (in this case, all the antihypertensive drugs whose dosage is increased), the task procedure will be processed. The task procedure specifies that, if the previous dose was in the therapeutic range of that drug, then the number of days that the drug was prescribed at the drug's previous dose must be compared to the period specified in the medical fact base as the minimum period before that drug's effect can be judged. If the period for which the drug was prescribed is shorter then this minimum time, then the critiquing statement "increased before effective" will be assigned to the pertinent event.

These task descriptions are represented as frames with three slots: a triggering event, a task procedure, and a critiquing statement. The execution of a task is performed by a program that processes such a frame. The program prepares a list that contains the pertinent events that matches the event description. Subsequently, the program executes the task procedure for each pertinent event individually. If the task procedure returns true, then the program assigns to the pertinent event the statement specified in the statement slot.

HyperCritic encodes the four types of tasks described in Section 3.3: preparation tasks, selection tasks, monitoring tasks and responding tasks. In the following four sections we will briefly describe how these abstract tasks are adapted for the hypertension domain.

4.4.5.1 Preparation Tasks

In HyperCritic, two types of preparation tasks have been implemented. These tasks hinge on the notion that certain pertinent events should be preceded by one or more preparations or observations. Executing these tasks involves tracing preparations associated with a particular event in the medical fact base. Subsequently, the program that executes the tasks verifies in the medical record whether these preparations indeed did take place. If they did not take place, then their absence is reported to the physician as a critique.

1. When drug therapy for hypertension is initiated by the physician, HyperCritic retrieves the minimum-workup requirements from the medical fact base. When a minimum-workup requirement is not available in the

patient description, then HyperCritic informs the physician of the absence of that minimum workup requirement.

2. For each drug that is started, HyperCritic retrieves the possible side effects of that drug. For each possible side effect, HyperCritic retrieves the monitoring requirements. If a baseline measurement is required, then HyperCritic evaluates whether the required measurement is available and is not out of date. If this measurement cannot be found in the patient description, then HyperCritic reports its absence.

4.4.5.2 Selection Tasks

Selection tasks embody the notion that situations exist that indicate that the physician's action or decision might not be the most appropriate. Executing these critiquing tasks involves retrieving from the medical fact base the situations in which the pertinent event is not the most appropriate, and examining in the medical record if these situations are present. HyperCritic can report the presence of seven classes of these situations.

1. For each drug that is started, HyperCritic retrieves the drug's contraindications from the medical fact base. Subsequently, HyperCritic determines whether these contraindications are present in the medical record. If a contraindication exists, then HyperCritic reports its presence.
2. For each drug that is started, HyperCritic retrieves the situations in which that drug would be an inappropriate choice from the medical fact base. Subsequently, HyperCritic determines whether any of these conditions are present in the patient description. In contrast to the previous task (screening for contraindications), this task consists of a search for situations in which the drug is not indicated, independent of the existence of contraindications. For example, if the physician prescribes *a short acting diuretic* to a patient with hypertension, HyperCritic will point out that this prescription constitutes an inappropriate selection since, for the treatment of hypertension, *long-acting diuretics* are the appropriate drugs. Hypertension, however, is not a contraindication for a short-acting diuretic.
3. The initial, minimum, and maximum doses of a drug are available in the medical fact base. For each antihypertensive drug that is prescribed, HyperCritic retrieves this dosage information, compares it to the actual

dose, and reports any conflicts.

4. The medical fact base includes the recommended increment of the dose of each drug. For each increment of dose, HyperCritic retrieves this information and determines whether the physician has increased the dose more rapidly or more slowly than is recommended. The presence of an increment outside the recommended range is reported.
5. The medical fact base contains knowledge concerning how long a drug has to be prescribed before that drug's effect can be judged. For each change in dose identified in the medical record, HyperCritic evaluates the duration for which the physician prescribed the drug at the previous dose. When the physician modifies the dose of a drug before its effect can be judged, HyperCritic reports that situation to the physician.
6. The medical fact base contains knowledge concerning the combinations of drugs that should be avoided. For each combination of drugs that the physician prescribes, HyperCritic checks whether the combination should be avoided. The presence of a combination that should be avoided is reported.
7. The classification of the World Health Organization for blood-pressure levels is available in the medical fact base. If the physician reduces the treatment for hypertension, HyperCritic evaluates whether the current blood-pressure levels are still elevated. HyperCritic comments on those situations in which the physician is decreasing the drug dose when elevated blood-pressure levels are still present. Similarly, HyperCritic comments on those situations in which the physician is increasing the drug's dose while the blood-pressure levels are judged acceptable.

4.4.5.3 Monitoring Tasks

Monitoring tasks represent the notion that a given pertinent event may require certain actions (for example, observing a particular parameter) at particular intervals. Executing these tasks involves tracing the monitoring requirements for a particular event, and subsequently verifying in the medical record whether these monitoring requirements have been met.

Monitoring requirements are recorded in the medical fact base. HyperCritic retrieves these requirements for each drug that is prescribed. If monitoring is required (for example, potassium levels must be monitored when the patient is treated with certain diuretics), then HyperCritic evaluates whether the required

measurements are available and not outdated. If these measurements cannot be found in the patient description, then HyperCritic reports their absence.

4.4.5.4 Responding Tasks

Responding tasks incorporate the notion that some finding related to a pertinent event may require a response. Executing these tasks involves tracing the situations to which a response will be required, and subsequently determining whether any of these situations is present in the medical record.

1. The adverse effects of drugs are recorded in the medical fact base. For each drug that is prescribed, HyperCritic retrieves the possible adverse effects of that drug and determines whether any of those effects are present in the patient description. If the effect is found, then HyperCritic reports its presence to the physician.
2. For each drug that the physician replaces with another drug, HyperCritic determines whether the patient had evidence of a side effect of the discontinued drug. If so, HyperCritic examines the medical fact base to determine whether the same side effect is also known to occur in the context of the newly prescribed drug. If it is, then HyperCritic reports both that the side effect is present and that the new drug may cause a similar side effect.
3. The suggested interval between visits is available in the medical fact base. HyperCritic evaluates whether the time period between the current visit and the previous visit is outside this range. If it is, HyperCritic reports the presence of this excessively long interval.

4.4.6 Critiquing Statements

During the execution of a task, if a task procedure evaluates true, HyperCritic will create a critiquing statement that subsequently is linked to the pertinent event that triggered the execution of that task. All possible critiquing statements are predefined: *possible side effect*, *dose is too high*, *medication is contraindicated*, and so on. When HyperCritic creates a critiquing statement for a pertinent event, the medical facts that were used are recorded. These medical

facts will be included in the output of the system.

4.4.7 Critique Generation

Based on the pertinent events, critiquing statements, and medical facts, HyperCritic generates its output by traversing augmented transition networks (ATN-s) [Miller and Rennels, 1988]. For each possible event description, critiquing statement, or entity in the medical fact base, an ATN is available that, when processed, generates descriptive text. HyperCritic first summarizes the current treatment, followed by a description of the critiquing statements generated by the preparation tasks. Subsequently, the system describes critiquing statements generated by the selection tasks, and the critiquing statements generated by the monitoring tasks and the responding tasks.

4.4.8 Current Status

HyperCritic has been written in an object-oriented fashion and is implemented on Xerox 1100-series workstations. HyperCritic currently is undergoing an extensive evaluation study in which the performance of the system is being compared to the performance of expert physicians using real patient data; the electronic medical records from ELIAS are retrieved from different sites in The Netherlands and are reviewed simultaneously by HyperCritic and by expert physicians. For the results of an initial evaluation, see [Van der Lei and Musen, 90]).

Figure 7 shows a fragment of an ELIAS medical record: The physician has discontinued a prescription for the beta blocking drug *metoprolol* and has started a prescription for *pindolol*, a beta blocker with properties somewhat different from those of metoprolol. The output of the system is as follows:

> *Treatment of hypertension at 5/10/89.*
>
> *You have stopped treatment with metoprolol and started treatment with pindolol in a dose of 15 mg per day.*
>
> *You have started treatment with pindolol. The recommended initial dose of pindolol is between 7.5 and 10 mg per day. You have started with a dose*

of 15 mg.

You have stopped treatment with metoprolol. The minimum time required before the effect of metoprolol can be judged is 14 days. The patient has taken the drug for only 11 days.

You have stopped treatment with metoprolol and started pindolol. Metoprolol may in rare instances cause depression. The diagnosis of agitated depression is present in this patient (5/10/89). Note that pindolol may cause a similar side effect.

HyperCritic first summarizes the current treatment, followed by three comments. All three comments first describe the pertinent event (the pertinent event that triggered the execution of the task; namely, starting pindolol, stopping metoprolol, and replacing metoprolol with pindolol), followed by a description of the medical facts used by the problem solver during the execution of that task (namely, the recommended initial dose, the minimum time required before the effect of the drug can be judged, and depression as a possible side effect of both metoprolol and pindolol). Next, the presence of a given situation (the high dose, the short treatment period, and the existence of a side effect) is pointed out.

A second example involves a 47-year-old male. During a routine check, a company physician discovers that the man's blood pressure is elevated. The patient is referred to his family physician. His automated-medical-record data are as shown here (see also Figure 6), followed by the output of HyperCritic. Not all data in the patient's chart are available in a coded format; coded data are shown in italics. The output of HyperCritic is also shown in italics.

3/30/89	Subjective:	During routine exam at work: elevated BP.
		No family history of high blood pressure.
		Does not smoke. No other risk factors.
	Objective:	*BP seated 165/105, left = right. Pulse 86, Weight 74 kg., Height 184 cm.*
		Physical exam: negative.
	Assessment:	*Raised BP, without diagnosis hypertension*
	Plan:	New appointment.

4/14/89	Objective:	*BP seated 165/100, left = right.*
4/28/89	Objective:	*BP seated 160/105, left = right.*
	Assessment:	*Hypertension, essential, with no end organ damage*
5/12/89	Objective:	*BP seated 160/110, left = right, pulse 86.*
	Plan:	Initiate workup.
5/15/89	Objective:	*ESR 6 mm, HCT 0.47, Hb 9.4 mmol/l, SGOT 6 U/l, SGPT 8 U/l, Gamma GT 21 U/l, K 4.1 mmol/l, Na 144 mmol/l, Creat 103 micrmol/l, urine: glucose negative, protein negative. ECG: no abnormalities.*
5/19/89	Objective:	*BP seated 165/100, Pulse 88.*
	Plan:	*Capoten 25 mg QD*
7/21/89	Subjective:	Patient complains about a cough. Family members similar cough.
	Objective:	*BP seated 155/1005*
	Plan:	*Capoten 25 mg TID* *Noscapine capsules 15 mg TID*

Output of HyperCritic on 5/19/89:

> *You have started treatment of hypertension with captopril in a dose of 25 mg per day.*
>
> *You have initiated the treatment of hypertension. This indicates that you have completed the workup of the patient. The minimum required workup includes an evaluation of the total serum cholesterol concentration. If elevated, further investigations (fasting levels of triglycerides and high-density lipoprotein) are required to determine both the nature of the elevated cholesterol concentration and the type of treatment (dietary measures, possibly drug treatment) required.*

This comment is generated by a preparation task (minimum-workup requirements). It reports the absence of a situation (that is, no total cholesterol is available).

Output of HyperCritic on 7/21/89:

> *You have increased the dosage of captopril from 25 to 75 mg per day.*

You have increased the dosage of captopril. If you prescribe 25 mg per day then the recommended next dose would be 50 mg per day. You are prescribing 75 mg per day.

You are prescribing captopril. An ACE-inhibitor may in rare occasions cause a decrease in renal function. It is therefore strongly recommended that you monitor the renal function. This evaluation should be performed about 4 weeks after the initiation of the treatment.

You are prescribing captopril. Captopril may cause a tickling cough. You have prescribed noscapine. It is recommended that you stop the treatment with captopril if you believe that tickling cough is a side effect of captopril.

The system first comments on the increment in the dosage of captopril; the presence of an excessive increment is reported. The second comment is a result of a monitoring task; the absence of a renal-function measurement is reported. The third comment is a result of a responding task; HyperCritic reports tickling cough as a possible side effect of captopril. This last comment also illustrates why we named the system HyperCritic: The physician probably knew about this side effect of captopril (she even recorded that family members also were coughing to emphasize the presumed viral cause of the patient's cough), yet the system, with its limited understanding of the reasoning underlying the physician's actions, failed to detect this fact. Even if information about the family also coughing were available in coded format, the system still would suggest that cough might be a side effect of captopril. HyperCritic does not model the notion of the *relevance* of a critiquing statement in the light of other findings.

Although languages such as CARE allow developers to create most if not all of the output of HyperCritic, the knowledge represented within HyperCritic includes additional levels of abstraction so that medical facts are separated from the knowledge used to generate critiques. We shall illustrate how these additional levels of abstraction can be used to drive knowledge acquisition and to provide support for the maintenance of the system.

4.5 KNOWLEDGE ACQUISITION

In Section 3, we argued that critiquing based on automated medical records requires, in addition to the interpretation of the medical record to determine the decisions and actions of the physician, two distinct types of knowledge: factual medical knowledge and critiquing knowledge. We identified generic classes of critiquing tasks and used this notion of abstract critiquing tasks to structure the medical knowledge encoded in the system.

Musen [1989c] provides a taxonomy for knowledge-acquisition tools that reflects a division based on the conceptual models that underlie these tools. This conceptual model determines the terms and relationships in which the user of one of these tools has to cast her thinking about a particular domain in order to use that tool. The taxonomy distinguishes (1) *symbol-based tools*, (2) *method-based tools*, and (3) *task-based tools*. The *symbol-based tools* force the user to express herself on the symbol level of a particular system -- for example, in terms of backward-chaining rules, or in the programming elements that constitute languages such as CARE. The *method-based tools* allow the expression of knowledge on the level of a particular domain-independent method for solving a problem (for example, heuristic classification [Clancey; 1985]); the user is shielded from the actual symbols used to implement that particular method. The user, instead, represents the task that the system is to perform in terms of an abstract model of the behaviors needed to solve certain classes of problems. Finally, the *task-based tools* concentrate on the general application tasks; they present the user with a set of terms and relations describing predefined general operations that can be performed in a given application area.

To illustrate the consequences of these different conceptual models underlying the knowledge-acquisition tools, we shall simulate how the same knowledge could be entered in three different systems: CARE, Essential-attending, and HyperCritic. We will specify that thiazide diuretics may cause a decrease in potassium level. We will also specify that, if the potassium becomes low during treatment with a thiazide diuretic, a warning should be given pointing out the decrease in the serum potassium, and implicating the thiazide as a possible cause. In addition, we want to state that, during treatment with thiazides, the potassium levels will have to be monitored at regular intervals: a baseline measurement when the therapy with thiazides is started, an evaluation 2 months after initiation of treatment, and subsequent yearly evaluations.

4.5.1 Knowledge Acquisition in CARE

In the absence of structures describing either the tasks that are performed by a system or the methods by which these tasks are realized, a knowledge-acquisition tool will be limited to the *symbol level* of that particular system. CARE does not have a model of the tasks performed or of the methods by which these tasks are effected. Consequently, CARE uses a symbol-based environment in which the elements that constitute the CARE language can be combined to make rules. Because CARE does not have abstract notions separating critiquing knowledge from medical knowledge, the system builder cannot define a side effect as an individual entity. Consequently, CARE cannot answer queries such as "What side effects of thiazide are known to the system?" With CARE, the system builder has to specify a specific situation that, if present, results in displaying a predefined paragraph. The detection of the low potassium level and the identification of thiazides as a possible cause requires the system builder to enter the following sequence of statements:

```
BEGIN block thiazides1
    IF          "K+ last" was LE 3.0
        AND "Thiazides use"
    THEN Patient is hypokalemic. This could be caused by thiazides.
    AND EXIT
END block thiazides1
```

That the potassium level has to be monitored during the use of thiazides requires the system builder to enter the following sequence of statements:

```
BEGIN block thiazides2
    IF          "Thiazides use" was on_after "most recent visit"
        AND (NO "K+ last" exists
            OR "K+ last" before 1 year ago)
    THEN  When starting a treatment with thiazides a baseline measurement
          of the potassium level should be obtained.
    AND EXIT
    IF          "Thiazides use"
        AND (NO "K+ last" exists
            OR "K+ last" was before 1 year ago V before "Thiazides first")
```

```
        AND "Thiazides first" was before 2 months ago
    THEN  Consider obtaining yearly "K+" to follow the effect
          of Thiazides.
    AND EXIT
END block thiazides2
```

This interaction requires that the user know exactly how the different symbols in the CARE language can be arranged to achieve the required behavior; the user has to express herself on the level of these CARE statements.

4.5.2 Knowledge Acquisition in Essential-attending

A well-known family of critiquing systems has been developed by Miller [1986]. One of those systems, HT-attending, critiques the management decisions of a physician's treatment of hypertension. A domain-independent shell, Essential-attending, allows the creation of new critiquing systems. Underlying Essential-attending is the ATN model, a hierarchical model, with different networks representing levels of decision and subdecision. Each ATN network consists of states connected by arcs. These arcs have associated conditions indicating when they should be used. Starting from the initial state of a network, the system traces a path from state to state along the arcs. The most important steps required to create a knowledge base involve the definition of the ATNs, the assignment of conditions to the arcs in the networks, and the definition of the prose that will be created when the network is traversed. The medical knowledge in the system is embedded in the hierarchical decomposition of the domain and in the conditions in the arcs of the networks.

Like CARE, Essential-attending does not contain abstractions representing either the tasks that are performed by the system, or the methods by which these tasks are addressed. Consequently, Essential-attending provides a *symbol-based* environment in which the elements that constitute the ATNs can be defined. The systembuilder has to understand the concept of an ATN and, subsequently, to formulate her domain (for example the treatment of hypertension) in the terms and relationships that are used in that model: decisions and sub-decisions, states, arcs, and conditions and comments associated with those arcs.

Unlike CARE, Essential-attending was not designed to rely for its data input on

automated medical records. To simplify our discussion, we assume that the data from the medical records can be parsed to the syntax Essential-attending requires. Essential-attending requires these structures to be triplets consisting of a name, an attribute, and a value.

One possible method for entering the knowledge about thiazides could be the following:

```
(DEFFRAME 'THIAZIDE-POTASSIUM-CRITIQUE '(
(IF  (AND (SAME AGENT THIAZIDES)
          (OR   (SAME PATIENT HYPOKALEMIC)
                (SAME POTASSIUM-MONITOR EXPIRED)
                (SAME POTASSIUM-BASELINE ABSENT)))
          COMMENT $THIA-INTRO SEQUENCE 1)
(IF (SAME PATIENT HYPOKALEMIC)
          COMMENT $THIA-HYPOKAL SEQUENCE 2)
(IF (SAME POTASSIUM-MONITOR EXPIRED)
          COMMENT $THIA-POTAS-MON SEQUENCE 3)
(IF (SAME POTASSIUM-BASELINE ABSENT)
          COMMENT $THIA-POTAS-BASE SEQUENCE 4)
) 'GENCOMMENTS)
```

Essential-attending refers to this as an *expressive frame*. The frame indicates that a critique should be generated when the patient is hypokalemic, or when the potassium levels require monitoring, or when a baseline measurement is absent. Essential-attending requires the definition of the corresponding prose comments:

```
(DEFPROSE '(
($THIA-INTRO     ((*para a thiazide diuretic may cause a decrease in serum
                 potassium *period) $POPTT T))
($POPTT          (*pop T T))
($THIA-HYPOKAL ((patient is hypokalemic *period this could be caused by
                 thiazides *period) $POPTT T))
($THIA-POTAS-MON      ((we recommend monitoring of the potassium levels at
                      regular intervals *period this evaluation should be done
                      two months after the initiation of treatment *comma
```

```
                    followed by a yearly evaluation *period) $POPTT T))
($THIA-POTAS-BASE   ((when a treatment with thiazides is initiated *comma
                    a baseline measurement of the potassium level should
                    be obtained *period) $POPTT T))
))
```

We have to specify the criteria for judging the potassium level, for judging whether a baseline measurement has been obtained, and for judging whether the potassium should be measured during treatment. Essential-attending provides the system builder the possibility for defining production rules. We define the following rules:

```
(SETQ THIAZIDE RULES '(
 (BASE-LINE RULE
     (IF  (AND    (SAME THIAZIDES STARTING)
                  (TEST POTASSIUM-MEASUREMENT VALUE ABSENT)))
     (THEN (POTASSIUM BASE-LINE ABSENT)))
 (MONITOR RULE
     (IF  (AND    (TEST OLDER-THAN THIAZIDES STARTING-DATE 60)
                  (OR (TEST POTASSIUM-MEASUREMENT VALUE ABSENT)
                      (TEST OLDER-THAN POTASSIUM-MEASUREMENT
                          DATE 366))))
     (THEN (POTASSIUM-MONITOR EXPIRED)))
 (HYPOKALEMIC RULE
     (IF (TEST LESS-THAN POTASSIUM-MEASUREMENT VALUE 3.0))
     (THEN (PATIENT HYPOKALEMIC)))
))
```

We assume that POTASSIUM-MEASUREMENT refers to the most recent potassium measurement, that POTASSIUM-MEASUREMENT VALUE refers to the result of the most recent potassium measurement, that POTASSIUM-MEASUREMENT DATE refers to the date the most recent potassium was obtained, that THIAZIDES STARTING indicates that the physician is starting a prescription for a thiazide diuretic, and that THIAZIDES STARTING-DATE refers to the first date thiazides were prescribed.

Note that the interaction with Essential-attending, like that with CARE, requires that the user know how the different symbols in Essential-attending must be arranged to achieve the required behavior. Note also that, because Essential-

attending does not have any abstract notions separating critiquing knowledge from medical knowledge, the system builder cannot define a side effect as an individual entity. Consequently, just like CARE, Essential-attending cannot answer queries such as "What drugs cause renal insufficiency as a side effect?"

4.5.3 Knowledge Acquisition in HyperCritic

Unlike the knowledge-acquisition tools that ask their users to describe an application in terms of the symbols required in the knowledge base, HyperCritic concentrates on the domain tasks and presents to the user the *terms and relations of a predefined task structure*. HyperCritic uses the notion of abstract critiquing tasks to structure the medical knowledge encoded in the system. The critiquing tasks require for their execution the presence of specific types of medical facts. For example, the task description that allows the system to search for possible side effects assumes the existence of information in the medical fact base that describes the side effects of drugs. Similarly, the monitoring tasks assume information that specifies when certain patient parameters should be monitored. In general, the tasks assume the presence and structure of the medical facts. Once the task structure has been implemented and the structure of the required medical facts has been defined, the system can be expanded by the addition of these medical facts. The development of a knowledge-acquisition module then involves the creation of an environment that allows the definition of these facts.

The manner in which developers enter medical knowledge into HyperCritic is shown in Figure 8. The interaction consists of using a mouse pointing device to make selections from menus. In the sample session shown in Figure 8, the user is entering that thiazides may cause a decrease in serum potassium levels. The user first indicates the type of changes that she wants to make in the knowledge base (the first four selections in Figure 8). Subsequently, the user selects the drug that is the cause of the side effect. The user then defines the side effect, together with an indication of the frequency and the clinical significance of the side effect. The user also provides an indication whether the existence of that side effect is well established (the existence of some effects can be uncertain in the sense that only isolated case reports are available in the literature). In addition, the physician may specify a premise

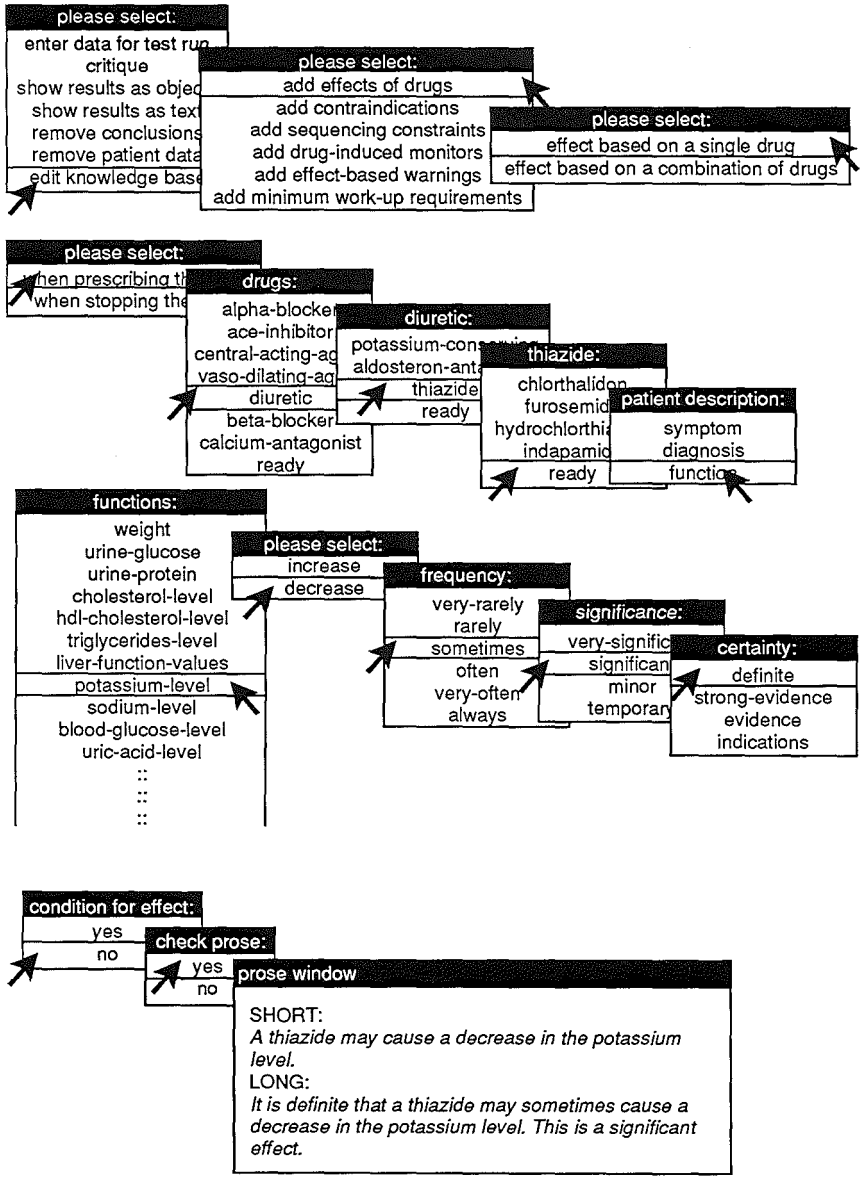

Figure 8: Adding a Side Effect to the Medical Fact Base
The interaction required to add knowledge to the medical fact base consists of using a mouse pointing device to make selections from menus. In this sample interaction, the user states that thiazide diuretics can cause a decrease in potassium levels. When the user selects "check prose," she activates the ATN that describes the side effect.

indicating conditions that must be present before the side effect is likely to occur (for example, the side effect is seen only when a given dose is exceeded). Once a side effect has been defined, the user may select the option "prose" (the final selection in Figure 8), which causes HyperCritic to traverse the ATN that generates the text describing the effect.

The user does not need to specify that the occurrence of the side effect should be reported. This critiquing knowledge (that side effects should be reported) is already captured by a task specification in HyperCritic that states that, for any antihypertensive drug that is prescribed at a given moment, all known side effects should be retrieved from the medical fact base. If any of these side effects is present in the patient description, the presence of that side effect should be reported.

Next, the user can associate monitoring activities with the newly entered side effect (Figure 9). The user selects the types of monitoring activities (the first three selections in Figure 9). In this case, the user specifies that the monitoring should be performed when a drug is being prescribed. The user then selects the drug that warrants the monitoring activity. Once the user has identified the drug, HyperCritic prepares a list of all known side effects of that particular drug. The user selects the side effect that requires monitoring, and specifies that the serum potassium level has to be evaluated 2 months after the start of the treatment, followed by a yearly evaluation. The user may select the option "prose" (the final selection in Figure 9), which causes HyperCritic to traverse the ATN that generates the text that describes the monitoring requirement.

If the user wishes to state that, prior to the initiation of a treatment with thiazides, a recent potassium evaluation should be available, then selecting the option "when starting the drug" (see Figure 9) provides a sequence of menus that would enable the definition of monitoring activities to be performed when the treatment is initiated. If the user wishes to specify the actions to be undertaken when the effect occurs, then the selection of the option "add effect-based warnings" allows the definition of those actions (for example, if the serum potassium level decreases during the use of thiazides, then the physician should consider adding a potassium-conserving diuretic).

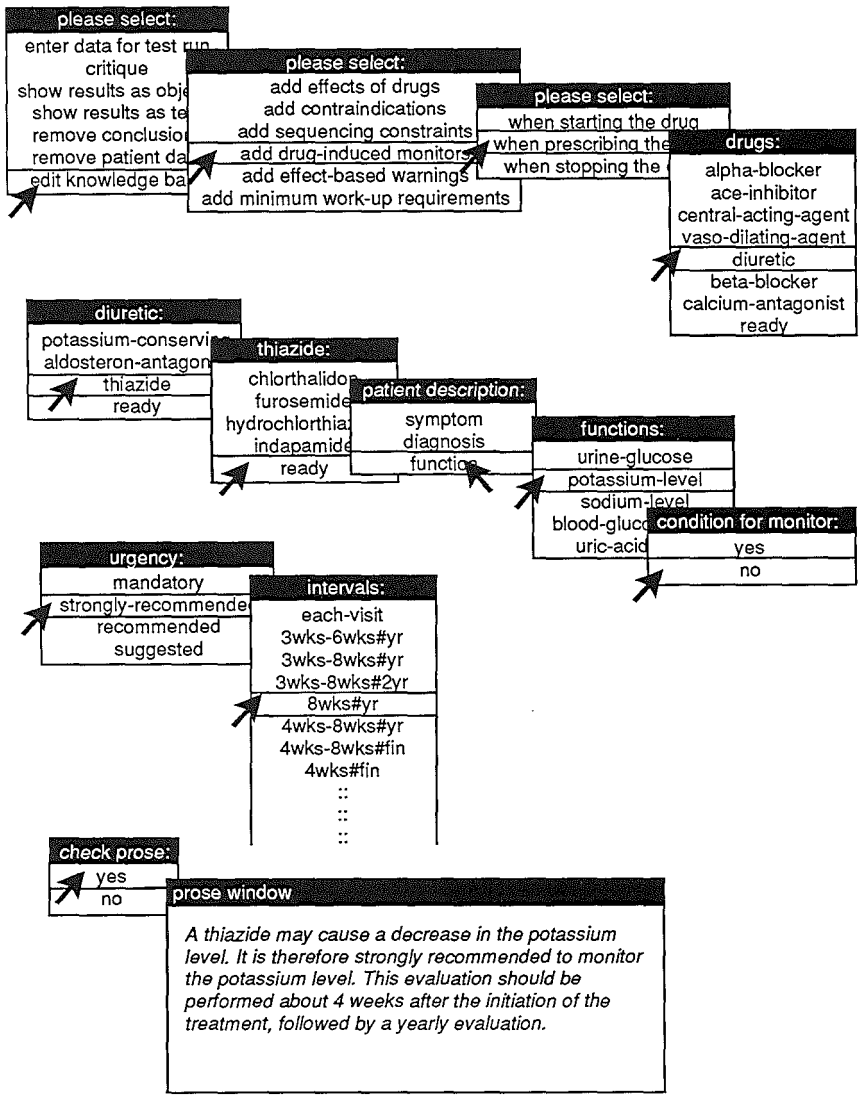

Figure 9: Adding Monitoring Requirements to the Medical Fact Base
The user adds to the medical fact base that, when thiazides are prescribed, periodic measurement of the serum potassium levels should be performed. When the user selects the option "check prose," she activates the ATN that describes the monitor.

We emphasize that the *task structure is not modified* by these interactions. The task structure ensures that the required behavior of the system is achieved whenever new facts are added to the medical fact base. Modifying the task structure is a complicated activity, and requires a thorough understanding of what the interactions among the tasks are and of how the structure of the medical fact base relates to these tasks.

4.6 SYSTEM MAINTENANCE

The separation between task structure and medical facts has advantages in the area of maintenance. Maintenance can be viewed on two levels: *maintenance of the task structure* and *maintenance of the medical facts*. As mentioned in the previous section, maintenance of the task structure is complex and requires an understanding of HyperCritic's formalism for representing tasks, of the manner in which classes of medical facts are defined, and of the manner in which queries to the medical fact based must be performed. On the other hand, the separation of critiquing knowledge from the medical fact base facilitates the maintenance of that medical fact base.

For example, suppose a particular drug has a given side effect. HyperCritic uses knowledge of this side effect in several decision procedures: screening for contraindications (when the drug is prescribed), suggesting clinical monitoring of patient functions (when a particular laboratory value is not available), suggesting the drug as a potential cause of an abnormality (when the system encounters an abnormal laboratory value), and avoiding a combination with another drug (when the combination of drugs is prescribed). In the CARE syntax, four different decision rules must be created to cover all these decision procedures. The side effect of the drug that is included in these rules would be embedded or implicit in the actual code; a system builder is unable to ask which CARE rules contain a reference to that particular effect. If the medical community subsequently decides, based on additional research, that a particular drug is not responsible for that side effect, then the system builder is forced to perform a search of the rules almost on a "character level," asking questions such as: "Which rules contain a string of characters that matches the drug name?"

The alternative we have chosen is to define the side effects of drugs in a medical fact base. Each side effect can be ascribed to entire classes of drugs. This approach prevents the error of adding a new drug in some class and forgetting to encode the side effect, since the side effect is inherited by all drugs in a given class. The system builder stores a side effect of a drug (or of a class of drugs) and, if knowledge of that effect is required elsewhere in the program, then an explicit reference to that side effect will be made. Consequently, when that particular side effect has to be changed, HyperCritic can produce graphic representations of those entities in the system that refer to this particular side effect (for example, monitors based on that side effect, or combinations of drugs involving that side effect). Moreover, when a system builder decides to remove this side effect from the medical fact base, HyperCritic will automatically remove all associated structures, such as monitors. This removal has no effect on the structure of the general critiquing tasks.

4.7 EVALUATION OF HYPERCRITIC

Any model is necessarily selective in what it contains. In creating models, the unusual properties of special cases are sacrificed to emphasize those of the general situation. For example, in a language such as CARE, developers can encode unusual knowledge by adding specific criteria to a rule. If the system builder wishes to suppress reporting the possibility of a particular side effect in some rare situation, she can easily include a criterion for suppressing such a report. HyperCritic, however, will report all putative side effects. The model does not support the definition of criteria for suppressing a particular report, as HyperCritic does not contain any model that defines when a critique is irrelevant. Although CARE does not contain an explicit model of the relevance of a comment, the flexibility of the CARE language allows programmers to adapt rules for this purpose in an ad hoc manner.

A model, ideally, identifies the most appropriate level of abstraction that will allow a particular task to be performed without introducing so much generality and subsequent rigidity that few actual tasks will fit the model. As soon as the application area does not "match" or "fit" the model, the system builder must either adapt the model or discard the model in its entirety [Musen and Van der Lei, 1989]. From the system-builders' viewpoint, it is important to understand the limitations of a given model. In this section, therefore, we will discuss the limitations of HyperCritic using data obtained from an evaluation study.

To evaluate the limitations of HyperCritic we performed an evaluation study that compared the performance of HyperCritic to the performance of human observers [Van der Lei and Musen, 1990]. In this study, the automated medical records of patients with hypertension were evaluated by eight physicians and by HyperCritic. If the majority of the physicians made a particular recommendation but HyperCritic did not make the same recommendation, we performed an analysis to establish the reasons that the system had failed to make that recommendation.

By design, HyperCritic will never judge a diagnostic investigation to be inappropriate based solely on the absence of an apparent indication, whereas physicians may judge a diagnostic investigation as superfluous if the rationale is not obvious from the medical record. This difference reflects a fundamental limitation of a system such as HyperCritic: For such a statement to be produced, an overall analysis of the patient is required. HyperCritic -- whose knowledge is limited to the domain of hypertension, and which has access to only that portion of the data in the computer-based medical record that is available in a coded format -- is unable to perform such an overall assessment. Human observers also frequently make errors in such an overall assessment of the patient based solely on an automated medical record. In a previous study [Van der Lei et al., 1989], we showed that the dominant reason that physicians believed a comment to be irrelevant was a misunderstanding of the physicians' intentions on the part of the persons who issued the critique.

Another reason that HyperCritic will fail to produce recommendations involves the inability of the system to interpret free text. In ELIAS, the majority of the patient-history data is available only in free text. Consequently, the ability to comment on the diagnostic workup and the selection of a drug is limited. ELIAS does not contain a coding scheme for the nondrug treatment of hypertension. Consequently, HyperCritic cannot provide any comments dealing with this important aspect of patient care. Moreover, the system cannot comment on the decision of the physician to change from nondrug treatment to drug treatment of hypertension.

In the absence of an absolute standard of care, we are confronted with a range of opinions that involve personal preferences, practice variation, and subjective

evaluation. HyperCritic embodies just one opinion in that range. In the evaluation study, HyperCritic tended to be more lenient (that is, less critical) than were the critiquing physicians. In our analysis, we estimated the sensitivity, specificity and predictive values of the recommendations made by individual physicians and HyperCritic by assuming that the majority opinion provided the final classification of a given recommendation as correct or incorrect (Table 3).

Table 3: Comparison of Sources of Comments [*Source:* Van der Lei and Musen, 1990, pg. 190, used with permission]

	Spec.	Sens.	PV Pos.	PV Neg.
Ph 1	0.82	0.72	0.91	0.54
Ph 2	0.78	0.70	0.89	0.50
Ph 3	0.77	0.74	0.89	0.54
Ph 4	0.75	0.65	0.87	0.46
Ph 5	0.70	0.86	0.88	0.66
Ph 6	0.69	0.73	0.86	0.50
Ph 7	0.52	0.88	0.82	0.63
Ph 8	0.36	0.94	0.79	0.70
HYPERCRITIC	0.88	0.74	0.94	0.57

The forgiving nature of HyperCritic is illustrated by the relatively low sensitivity achieved by the system (see Table 3). HyperCritic provides decision support as a byproduct of routine data-management activities. The physician does not specifically request advice. Therefore, one of the goals and challenges is to avoid generating excessive numbers of comments, particularly since the false-positive advisory reports may generate antagonistic responses. Consequently, for systems such as HyperCritic, ensuring a high specificity is of great importance. The dilemma, of course, is that increasing the sensitivity will, in all likelihood, decrease the specificity.

The evaluation study also showed areas in which the task structure needs to be enhanced. One such area is the model for searching for possible secondary causes of hypertension. In the current version of HyperCritic, searching for causes of hypertension is limited to the diagnostic investigations required by a minimum workup. Consequently, if all findings in the minimum workup are normal, the system assumes that the diagnosis of essential hypertension is correct. For example, one patient represented in the data set had sustained a severe blow on the head resulting in a cerebral contusion. This condition may

cause a period of high blood-pressure readings. HyperCritic failed to notice this possible cause of the hypertension, as the program contains a model of a workup and not a model of the possible causes of hypertension.

The frequency of comments generated by the system, however, indicates that there are deficiencies in the management of hypertensive patients, and that both human observers and HyperCritic can provide useful comments to physicians. Although the results of this evaluation are promising, a field evaluation of the system that includes feedback to the treating physician will be performed to assess the effect of HyperCritic on the delivery of care.

4.8 DISCUSSION

HyperCritic constitutes an architecture for generating critiques of patient-therapy, based solely on data from automated medical records. In the HyperCritic program, we have explored the use of this approach in the hypertension domain. We have shown that critiquing knowledge can be separated from medical knowledge. Moreover, we have shown that the critiquing knowledge can be represented as a small set of generic critiquing tasks. These critiquing tasks provide the structure for the medical knowledge base. The explicit separation of critiquing knowledge from medical knowledge enables us to distinguish between acquisition of the critiquing knowledge and acquisition of the medical knowledge, and also between the maintenance of the task structure and the maintenance of the specific facts in the medical knowledge base.

4.8.1 Task-Based Architectures

The separate modeling of control knowledge and of content knowledge to provide a basis for knowledge acquisition and maintenance of large decision-support systems has been an active area of research in recent years [Clancey, 1983; Chandrasekaran, 1986; McDermot, 1986; Neches et al., 1985]. Experience with early decision-support systems has shown the complications that arise if control and content knowledge are intermixed. For example, XCON is a very large rule-based system used routinely at Digital Equipment Corporation for the configuration of computer equipment. Although not

developed for a medical application area, XCON demonstrates clearly that conventional approaches to the development of large expert systems may lead to significant maintenance problems for the systems' developers. The original XCON system contained approximately 250 production rules, and grew to over 3300 rules by 1983 [Bachant and McDermott, 1984]. Bachant and McDermott [1984: page 28] observed that, before the system was one year old, "the incremental addition of knowledge resulted in a significant amount of redundancy and a penchant for *ad hocery*. To the extent that adding knowledge to the system involves human intervention, the general lack of cleanliness and conciseness provides an obstacle to the system's further development." To address this overwhelming maintenance problem, workers at Digital currently are imposing a task structure on the XCON rule base by using a problem-space approach of the RIME programming methodology [Soloway et al., 1987].

The notion of separating control knowledge from content knowledge also has been an important concern for developers of large medical knowledge bases. When Clancey [1987a] attempted to use the MYCIN knowledge base for tutoring medical students, he discovered that the knowledge required for tutoring was not present in the original MYCIN knowledge base. Subsequently, Clancey created a separate knowledge base of tutoring strategies when he developed GUIDON [Clancey, 1987a]. The result, however, was unsatisfactory because control knowledge in MYCIN could not be separated from medical content knowledge. The GUIDON program itself could not distinguish an entry in the MYCIN knowledge base that was a medical fact from an entry that was added simply to control MYCIN's inference behavior. Clancey thus built NEOMYCIN in an attempt to separate out MYCIN's control knowledge from program's medical content knowledge [Clancey and Letsinger, 1981]. In NEOMYCIN, the domain-independent control strategy is captured in the form of meta-rules that define an abstract diagnostic method. Based on an analysis of both these meta-rules and of other expert systems, Clancey further abstracted NEOMYCIN's diagnostic strategy and defined the program's problem-solving behavior in terms of heuristic classification [Clancey, 1985]. The heuristic classification model now forms the foundation of a special expert-system shell called HERACLES [Clancey, 1987b].

In recent years, builders of medical expert systems have placed particular emphasis on the development of environments that can distinguish control strategies from medical content knowledge [Chandrasekaran, 1986; Musen,

1989b]. Analogous task abstractions for critiquing systems, however, have not been developed; to our knowledge, HyperCritic is the first critiquing program that uses a task-based architecture. Although Essential-Attending is a development tool that allows users to build critiquing systems, Miller [1986, page 107] points out, however, that there is nothing inherently medical or inherently "critiquing" about Essential-attending. As shown in section 5.2, Essential-Attending intermixes critiquing knowledge with medical content knowledge. Similarly, CARE [McDonald, 1981] does not support the separation of critiquing knowledge from medical content knowledge. Consequently, knowledge acquisition and maintenance have to be performed at the symbol level in both these systems. The experience with decision support systems such as XCON, however, shows that knowledge acquisition and maintenance on this level may not be sufficient for the creation and maintenance of larger knowledge bases.

In addition to advantages in the area of knowledge acquisition and maintenance, the explicit separation between critiquing knowledge and medical knowledge provides opportunities for re-using both the task structure and the associated medical content. The need to avoid redundancy is critical, particularly in larger knowledge bases. Fox [1987], for example, has estimated that a knowledge-based system designed to satisfy the information needs of British general practitioners would require at least a million different medical facts. Any steps that can be taken to eliminate redundancy in a knowledge base of this size will have important consequences for system performance. Fox, who hopes that his Oxford System of Medicine (OSM) may one day contain a knowledge base that approaches this size, recognizes the importance to distinguish between "symbolic decision procedures" on one hand, and "medical facts" on the other [Fox, 1989]; the former represent the tasks that OSM must perform, whereas the latter comprise the specific medical facts with which to perform those tasks. Fox points out that it is not surprising that re-using medical knowledge in different situations has met with only limited success: Using the same knowledge in a different situation requires the separation of those components that describe the purpose (when and how this knowledge is to be used) from those components that are independent of purpose. If the description of the tasks is tightly interwoven with other knowledge, then separation will be impossible -- limiting the use of the knowledge to the particular purpose the designers had in mind when they

created the system, and causing redundancy in the knowledge base when the same medical facts are to be applied in different settings. Furthermore, when the factual knowledge needs to be updated, there is no guarantee that system maintainers will perform the required updates in a consistent manner in all places where the knowledge is duplicated.

Musen [1989b] has addressed the issue of re-usability of knowledge in the context of work on medical knowledge acquisition. Adopting the terminology used by developers of database models, Musen makes a distinction between the description of the decision model -- the intention -- and the specific facts required by that model -- the extension. Musen has shown in the PROTÉGÉ program how a decision model can be readily re-used by automatically generating custom-tailored editors that physicians can use to enter appropriate extension of that model. PROTÉGÉ helps its users to build models for tasks that can be solved by successive refinement of skeletal plans [Friedland and Iwasaki, 1986] (for example, the administration of cancer therapy). PROTÉGÉ then produces special-purpose, graphical tools with which expert physicians define the details for that model (for example, the administration of a particular cancer-therapy plan). The knowledge bases of systems created using PROTÉGÉ -- unlike those of both HyperCritic and the Oxford System of Medicine -- do not contain any facts that are not dependent on specific decision procedures, as PROTÉGÉ does not allow entry of facts without reference to the context in which the facts will be used during problem solving. HyperCritic and OSM, however, both allow the user to enter content knowledge that might never be used in any problem-solving procedure.

In the HyperCritic system, we model the process of critiquing the physician's therapy of hypertension based on data in an automated medical record. Our underlying assumption is that the critiquing knowledge can be separated from the medical knowledge, and that critiquing knowledge can be modeled as a limited set of general critiquing tasks. If this assumption is correct, than the critiquing tasks we have identified can be re-used in other domains -- for example, in the treatment of hyperlipidemia, congestive heart failure, or rheumatoid arthritis. Moreover, the medical fact base can be re-used. For example, if similar drugs are used in different domains, the side effects and monitoring requirements need not be duplicated in the knowledge base.

4.8.2 The Utility of HyperCritic's Critiques

The HyperCritic architecture assumes that all inputs to the system will originate from an electronic medical-record system. As with systems such as CARE and HELP, the requirement that physicians must interact with the computer directly to obtain decision support is therefore obviated -- although physicians routinely enter medical data into ELIAS when maintaining their medical records. The generation of critiques by HyperCritic occurs without the physicians' explicitly requesting a critique, and in a manner that in no way alters the physicians' daily routine. We believe that the ability of HyperCritic to generate comments in a transparent manner will enhance acceptability of the system. We recognize, however, that the restricted nature of the input to the system limits both the scope and the applicability of the comments that HyperCritic can generate. Given the design decision to omit several classes of critiquing tasks from the HyperCritic knowledge base, we have attempted to maximize the likelihood that the system's comments are germane, rather than to maximize the number of comments that the system can generate. A major problem faced by all automated systems that offer advisories on the basis of medical-record data, however, is assuring that only relevant comments are issued.

Currently, HyperCritic does not model the potential relevance of its comments. Because a comment's relevance depends on the personal preferences of the physician to whom that comment is presented, HyperCritic requires a mechanism for detecting those comments that the physician repeatedly ignores. Further work on HyperCritic will concern the development of an explicit model of the physician who receives the critiques. Based on this user model and the critiquing task structure, HyperCritic will plan its discourse with the user.

In a previous study [Van der Lei et al., 1989], we evaluated the response of physicians to critiques generated by human observers. We identified four major reasons why a critique might be judged irrelevant. First, it is easy for a physician to misunderstand the intentions of a colleague based solely on the data obtained from the medical record. The medical record is primarily a record of the actions performed; physicians tend to record only their actions, rather than the rationale for those actions [Weed, 1971]. A particular prescription may not be labeled "for the treatment of hypertension". Diagnostic tests may not be labeled "to exclude disease X". The underlying reasoning has to be

reconstructed. Moreover, not all actions or decisions of the GP are mentioned in the medical record. Missing data may lead to an incorrect interpretation of other data, and subsequently may lead to the generation of an irrelevant critique. (The word *irrelevant* denotes a situation in which the treating physician identifies the critiquing physician's incorrect assumptions, recognizes the consequences of these incorrect assumptions, and subsequently disregards the critiquing physician's advice.)

A correct understanding of the intentions of the treating physician, however, does not guarantee that the critique will be judged relevant. Clinical practice is not solely based on scientific facts and evidence. Both training and practice setting have an effect on medical decision-making [Petersdorf, 1978; McDonald, 1981; Palchik et al., 1987]. When a physician receives a critique, he will evaluate whether, in his practice setting and with his training, the critique is relevant. In our previous study, the three main reasons physicians gave for judging a comment less than relevant were: (1) the physician disagreed with the medical reasoning; (2) the physician agreed with the principle but he would prefer to modify the recommendation to suit his practice setting; or (3) the physician felt that the advice had no consequence for the decision he had to make [Van der Lei et al., 1989].

When the physician judges the advice of a peer to be less relevant or irrelevant because he disagrees with the medical reasoning, he is indicating that he recognizes the inferences of his peer, and disagrees with those inferences. The resulting advice subsequently is judged irrelevant. The physician does not question whether his peer understood the treating physician's original intentions; rather, the physician simply disagrees with his peer's presumed line of reasoning. When the physicians states, however, that the advice of a peer has no consequences for the treatment of the patient, the physician does not disagree with the inferences made by a peer; instead, the physician states that the comment can have no effect on his decision making. When reviewing the advice of a peer, the physician may be able to distinguish general treatment principles from the translation of these principles into specific actions. The physician may agree that the principles should be applied, but may disagree with the actions that his peer recommends. The treating physician's opinion thus differs from that of his peer when viewed at the level of specific actions to be taken, yet the opinions may well be congruent at the more abstract level of treatment principles. Developing mechanisms for HyperCritic to detect such distinctions and to tailor its comments accordingly will be subject of future

work.

Currently, HyperCritic will not modify its discourse based on the responses of the user to the program's previous suggestions. To adapt its comments, HyperCritic (1) will have to contain knowledge about the reasons the physician may have for ignoring advice, and (2) will have to deduce the reasons that apply in particular cases from the primary data in the medical record.

4.8.3 Critiquing at the Knowledge Level

Our work has identified four types of domain-independent critiquing tasks (preparation tasks, selection tasks, monitoring tasks, and responding tasks), which can be distinguished from the medical content knowledge that a critiquing program uses to generate comments concerning a physician's therapy. Our model of critiquing not only has simplified development and maintenance of the HyperCritic system for commenting on hypertension therapy, but also will facilitate both our future extensions to HyperCritic critiquing mechanisms and adaptation of the system for other medical application areas.

Builders of other medical decision-support systems that use medical-record data as their primary input have recognized the difficulties of system development and the limited degree to which knowledge is reusable [Clayton et al., 1989]. Several workers in medical informatics recently have suggested that creation of a standard syntax for representing knowledge would facilitate knowledge re-use, by allowing knowledge representation within applications to be consistent, and by allowing system builders to share knowledge across applications. Although such efforts are important, attention to symbol-level issues of knowledge representation is not sufficient to allow one developer to import the knowledge bases created by other workers, or to allow adaptation of one knowledge base for another purpose. The re-use of encoded knowledge demands attention to knowledge-level issues as well. HyperCritic defines abstract critiquing tasks at the knowledge level, permitting system builders to understand explicitly the stereotypic *behaviors* generated by the system in producing a critique. More important than the re-use of knowledge-base symbols, HyperCritic makes it possible for developers to re-use these critiquing behaviors in the context of additional domain knowledge, and in the context of

entirely new domains.

Previous medical critiquing systems (for example, HELP, CARE, and ATTENDING) were built from symbol-level primitives. Writing HELP-sectors, CARE rules, and expressive frames for ATTENDING requires the developer to create the appropriate programs and data structures that will generate the required critiquing behavior. Critiquing programs such as CARE and ATTENDING have no explicit knowledge of the critiquing process itself; knowledge of what should trigger a critique, of what a critique includes, and of how a critique should be presented to the user is hidden in the rules and expressive frames. This knowledge of the critiquing process becomes manifest only when the actions of the program are observed. HyperCritic, on the other hand, elucidates the process of critiquing a physician's therapy as a distinct phenomenon to be modeled -- independent of the content knowledge on which the critiques are based. By helping to clarify the critiquing process itself, HyperCritic will allow us to investigate variations on the delivery of computer-based advice, and to explore how computers can best influence physician decision making.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the Netherlands Heart Foundation, grant 88.236. We are indebted to our clinical collaborators for their advise and support, especially R. Frans Westerman of the Department of Medicine at the Free University in Amsterdam. We are especially grateful to our colleagues Jan H. van Bemmel, Terry F. Blaschke, Joop S. Duisterhout, Wilfried M. Boon, Astrid van Ginneken, and Jan Kors, who provided valuable comments on previous drafts of this paper. We thank Mees Mosseveld for competent programming assistance and Rosa J.J.R. Scholte for secretarial support.

REFERENCES

Barnett GO, Cimino JJ, Hupp JA, et al. (1987). DXplain: An evolving diagnosis decision-support system. *Journal of the American Medical Association*, 1987;258:67-74.

Bachant J, McDermot J (1984). R1: Four years in the trenches. *AI Magazine*, 1984:

Chandrasekaran B (1986). Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE Expert*, 1986;1:23-30.

Clancey WJ, Letsinger R (1981). NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching. In: *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 829-36, Vancouver, Canada.

Clancey WJ (1983). The advantages of abstract control knowledge in expert system design. In: *Proceedings AAAI-83*, pages 74-78. American Association for Artificial Intelligence, Morgan Kaufman Publishers.

Clancey WJ (1985). Heuristic classification. *Artificial Intelligence*, 1985;27:289-350.

Clancey WJ (1987a). *Knowledge-Based Tutoring: The GUIDON Program.* Cambridge, MIT Press.

Clancey WJ (1987b). From Guidon to Neomycin and Heracles in twenty short lessons. In: Vanlamsweerde A ed., *Current Issues in Expert Systems*, pages 79-123. Academic Press, London.

Clayton PD, Pryor TA, Wigertz OB et al. Issues and structures for sharing medical knowledge among decision-making systems: the 1989 Arden Homestead Retreat. In: *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care*, pages 116-21, Washington DC. IEEE Computer Society Press.

Evans RS, Larsen RA, Burke JP, et al. (1986). Computer surveillance of hospital-acquired infections and antibiotic use. *Journal of the American Medical Association*, 1986;256:1007-11.

Friedland PE, Iwasaki Y (1985). The concept and implementation of skeletal plans. *Journal of Automated Reasoning*, 1985;1:161-208.

Fox J, Glowinski A, O'Neil M. (1987) The Oxford System of Medicine: A prototype system for primary care. In: *Proceedings of the European Conference on Artificial Intelligence in Medicine*, pages 213-16, Marseilles, France. Springer Verlag, Berlin.

Fox J (1989). Symbolic decision procedures for knowledge based systems. In: Adell H ed., *The Handbook of Knowledge Engineering*. McGraw-Hill, New York.

Glowinski A, O'Neil M, Fox J (1989). Design of a generic information system and its application to primary care. In: *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 221-33, London, United Kingdom. Springer-Verlag, Berlin.

Gross F, Pisa Z, Strasser T, et al. (1984). *Management of Arterial Hypertension.* World Health Organization, Geneva, Switzerland.

Gruber TR (1988). Acquiring strategic knowledge from experts. *International Journal of Man-Machine Studies*, 1988;29:579-97.

Hasling DW, Clancey WJ, Rennels GD (1984). Strategic explanations for a diagnostic consultation system. In: Coombs MJ ed. *Developments in Expert Systems*, pages 117-33. Academic Press, New York.

Lamberts H, Wood M eds (1987). *International Classification of Primary Care.* Oxford University Press, Oxford.

Langlotz CP, Shortliffe EH (1983). Adapting a consultation system to critique user plans. *International Journal of Man-Machine Studies*, 1983;19;479-96.

Lanzola G, Stefanelli M, Barosi G, et al. (1989). A knowledge system architecture for diagnostic reasoning. In: *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 234-47, London, United Kingdom. Springer-Verlag, Berlin.

McDermott J (1986). Making expert systems explicit. In: Kugler H. ed. *Information Processing 86*, pages 539-544, Dublin, Ireland. International Federation of Information Processing Societies, Elsevier Science Publishers B.V. (North Holland).

McDonald CJ (1981). *Action-Oriented Decisions in Ambulatory Medicine.* Year Book Medical Publishers, Chicago.

McDonald CJ, Hui SL, Smith DM, et al. (1984). Reminders to physicians from an introspective computer medical record. *Annals of Internal Medicine*, 1984;100:130-8.

Miller PL (1986). *Expert Critiquing Systems: Practise-Based Medical Consultation by Computer.* Springer-Verlag, New York.

Miller PL, Rennels GD (1988). Prose generation from expert systems, an applied computational linguistics approach. *AI Magazine*, 1988;9:37-44.

Miller RA, Masarie FE (1990). The demise of the "Greek Oracle" model for medical diagnosis systems. *Methods of Information in Medicine*, 1990;29;1-2.

Musen MA, Van der Lei J (1989). Knowledge engineering for clinical consultation programs: modeling the application area. *Methods of Information in Medicine*, 1989;28:28-35.

Musen MA (1989a). The strained quality of medical data. *Methods of Information in Medicine*, 1989;28:123-5.

Musen MA (1989b). Automated support for building and extending expert models. *Machine Learning*, 1989;4:349-77.

Musen MA (1989c). Conceptual models of interactive knowledge acquisition tools. *Knowledge Acquisition*, 1989;1:73-88.

Neches R, Swartout WR and Moor J (1985). Explainable and maintainable expert systems. In: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 382-9, Los Angeles, California.

Newell A (1982). The knowledge level. *Artificial Intelligence*, 1982;18:87-127.

Palchik NS, Dielman TE, Woolooscroft JO et al. (1978). Practice preferences of primary care and traditional internal medicine house officers. *Medical Education*, 1987:21:441-9.

Petersdorf RG (1978). The doctor's dilemma. *New England Journal of Medicine*, 1978;299:628-34.

Pryor TA, Gardner RM, Clayton PD, et al. (1983). The HELP system. *Journal of Medical Systems*, 1983;7:87-102.

Royal College of General Practitioners (1984). *Classification of Diseases, Problems and Procedures.* Occasional paper 26, London.

Shortliffe EH (1987). Computer programs to support clinical decision making. *Journal of the American Medical Association*, 1987;258:61-6.

Soloway E, Bachant J and Jensen K (1987). Assessing the maintainability of XCON-in-RIME: Coping with the problems of a very larger rule-base. In: *Proceedings AAAI-87*, pages 13-7, Seattle, Washington.

Swartout WR (1981). *Producing Explanations and Justifications of Expert Consulting Programs.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts. MIT/LCS/TR-251.

Tu SW, Kahn MG, Musen MA, et al. (1989). Episodic skeletal-plan refinement based on temporal data. *Communications of the ACM*, 1989;32:1439-55.

Van der Lei J, Van der Heijden P, and Boon WM (1989). Critiquing expert critiques. Issues for the development of computer-based monitoring in primary care. In: *Proceedings of MEDINFO 89*, pages 106-10, Singapore. North-Holland Publ Comp, Amsterdam.

Van der Lei J and Musen MA (1990). Critiquing physician decision making using data from automated medical records: An evaluation of Hypercritic. In: *Proceedings of the AAAI 1990 Spring Symposium on Artificial Intelligence in Medicine*, pp. 184-9, Stanford, California.

Warner HR (1978). *Computer-Assisted Medical Decision Making.* Academic Press Inc, New York.

Westerhof HP, Boon WM, Cromme PVM, et al. (1987). ELIAS: Support of the Dutch General Practitioner. In: Reichertz PL, Engelbrecht R, Picollo U, eds. *Present Status of Computer Support in Ambulatory Care*, pages 1-10. Springer-Verlag, New York.

Weed LL (1971). *Medical Records, Medical Education, and Patient Care: The Problem-Oriented Record as a Basic Tool.* Case Western Reserve Press, Cleveland.

CHAPTER 5

Review of Physician Decision Making using Data from Computer-stored Medical Records

Submitted for publication

Johan van der Lei, Mark A. Musen, Emiel van der Does,
Arie J. Man in 't Veld, Jan H. van Bemmel

ABSTRACT

Background. Many researchers in medical informatics are developing computer-based medical-record systems. Automated review of these medical records is expected both to limit errors in the delivery of care and to control costs. We investigated whether computer-based medical records contained sufficient information to generate useful critiques, and compared the limitations of computer-based critiquing with those of peer review.

Methods. We had a panel of eight physicians and a computer-based critiquing system known as HyperCritic review 20 automated medical records (comprising a total of 243 visits) for patients undergoing treatment of hypertension. We used the majority opinion of the reviewing panel as the yardstick in determining the appropriateness of the review.

Results. We received a total of 468 comments concerning patient management. Two-hundred-and-sixty comments were judged correct by 6 or more of the 8 physicians. Of these 260 comments, the critiquing system reproduced 118. The main reasons that the system did not produce the other 142 comments were (1) insufficient data in the computer-based medical record, (2) absence of sufficient medical consensus, and (3) omissions in the knowledge base of the critiquing system. When compared to individual members of the panel, however, the computer system performed better (*Index of Merit* 0.62) in its limited domain than did peer reviewers (*Index of Merit* ranging from 0.30 to 0.56).

Conclusions. We conclude that automated review of computer-based medical records in a limited domain can compete successfully with peer review. Further development of computer-based review requires that two main tasks be addressed: the development of a computer-based medical record that captures the reasoning pattern of the physician, and the development of widely accepted practice guidelines.

5.1 INTRODUCTION

In recent years, researchers have developed several systems that provide physicians with computer-based medical records. Some systems use hybrid computer- and paper-based versions of the medical record, whereas other systems provide the physician with a "paperless" patient chart. Three benefits accrue from the use of computer-stored medical records: (1) improved logistics and organization of the medical record speeds delivery of care and improves caregivers' efficiency, (2) automatic computer review of the medical record limits errors and controls costs, and (3) systematic analysis of past clinical experience guides future practices and policies [1].

Parallel to the development of computer-stored medical records, various decision-support systems have been designed and fielded [2]. During the 1970s, the medical-informatics community was particularly optimistic and enthusiastic about the future of computer-assisted decision making. Encouraged by systems that performed at an expert level [3,4], researchers had high expectations that artificial-intelligence and expert consultant systems might play an important role in medical practice [5-7]. Widespread use of these systems has not materialized, however, and initial optimism has been tempered by realism. Miller and Masarie [5] argue that one of the fundamental causes of this lack of success is the ambitious model for decision support that early developers incorporated in their systems. Miller and Masarie refer to this model as the *Greek oracle*. The physician, unable to cope with a given medical problem, would submit all relevant patient information to the system. The oracle would then solve the problem, with the physician as a passive observer who waited for the oracle to reveal its ultimate verdict. Miller and Masarie keenly observe that developers should remember that the most useful intellect to be brought to bear during a consultative session is that of the physician-user [5].

An alternative to the Greek oracle model is the *critiquing model*. Here, the physician submits to the program, in addition to patient-specific data, the decisions he intends to make. The program evaluates these decisions and expresses agreement or suggests alternatives [8]. For example, the critiquing system HT-Advisor [9] was available on the electronic-communication network of the American Medical Association. Physicians could dial in, and could submit to the program cases of patients who had hypertension, with a description of the antihypertensive therapy that the physicians intended to prescribe. HT-Advisor would analyze the proposed treatment and either would approve or

would suggest modifications in the treatment.

It seems obvious, at first glance, that the critiquing model for providing computer-based decision support and the use of computer-based medical records can be integrated: By electronic transfer, the data in the automated medical records can be sent to the critiquing system. The critiquing program evaluates the decisions of the physician and, if appropriate, suggests reasoned alternatives. From the viewpoint of the physician, he enters data in a computer-stored medical record, but, behind the scenes, the medical-record system forwards the patient data to a critiquing program.

Collaborations thus are developing between workers in computer-stored medical records and in computer-based decision support [1,2,10]. Using medical data collected for one purpose in the context of another purpose, however, presents problems [11]. For example, when medical data collected for billing purposes are used for epidemiological studies, they need to be interpreted in the context of how the financial consequences of a given diagnosis may have affected the prevalence of particular diagnoses [12]. Similarly, computer-stored medical records are not designed explicitly as data-entry modules for critiquing systems. If we are to use data from computer-stored medical records as input to critiquing systems, we must first analyze whether the data are suitable fot that purpose.

We investigated the possibilities and limitations of automated critiquing on the basis of data from computer-based medical records by comparing the performance of human observers with that of a computer-based critiquing system. Our goal was to assess the appropriateness of comments offered by a panel of peer reviewers and by the HyperCritic computer system for the critique of hypertension management [13], when the data for review were limited to those obtainable from ELIAS, a commercially available medical-record system [14].

5.2 PATIENTS AND METHODS

We had eight physicians and an automated critiquing system review automated patient records; direct interaction with the physician who treated the patients was not allowed. We compared the comments made by the physicians with those made by the critiquing system. We first describe the medical record

system (Section 5.2.1) and the critiquing system (Section 5.2.2). In Section 5.2.3, we describe the patients whose medical records were reviewed; and in Sections 5.2.4 through 5.2.6, we describe our study design.

5.2.1 Medical Record System

ELIAS is an information system designed for use by general practitioners (GPs) [14]. The system is commercially available in The Netherlands, and is used at over 200 sites. ELIAS supports a wide range of capabilities (for example, patient scheduling, financial administration, medical record keeping) and provides the GP with a paperless office; GPs using ELIAS typically no longer maintain a paper-based medical record. A video-display unit is located on the desk of the physician, who enters patient data via keyboard input. At some sites, ELIAS is integrated with computer facilities of other health-care institutions in the region (for example, hospitals and laboratories), providing electronic-communication facilities to other health-care workers [15]. Updating of the computer-based medical record can be performed automatically in response to patient-specific electronic mail (for example, laboratory reports and data from hospital admissions that are transferred by electronic data interchange (EDI) from one system to another).

Although not all data in ELIAS are coded, laboratory data and prescriptions are always stored in a machine-interpretable format. Vital-sign measurements obtained during physical examination (blood pressure, pulse rate, and so on) also are coded. A complete database of all drugs available in The Netherlands is on-line. One of two diagnosis-coding schemes may be used at a given ELIAS site: the *International Classification of Primary Care* (ICPC) [16] or the *Classification of Diseases, Problems and Procedures* of the Royal College of General Practitioners [17]. ELIAS supports a problem-oriented record-entry system, although use of the problem-oriented approach is not mandatory.

5.2.2 Critiquing System

Our laboratory has developed a system, called HyperCritic, that critiques a physician's treatment of hypertensive patients [13]. HyperCritic takes as input the electronic ELIAS medical record for a single patient; it will not request any data other than those available in this medical record. The output of HyperCritic

is in the form of text. HyperCritic does not reside on the same computer as does ELIAS; HyperCritic can be reached, however, by electronic-communication facilities. The system is the subject of ongoing research in our laboratory, and has not yet been released for routine use.

HyperCritic offers comments concerning drug therapy and laboratory tests administered to patients who have benign, essential hypertension. HyperCritic itself does not attempt to assign the diagnosis of hypertension based on primary data in the patient's ELIAS chart. Rather, the system critiques a physician's therapy of hypertension only after the physician has made that diagnosis. Furthermore, HyperCritic does not judge any diagnostic investigation as inappropriate based solely on the absence of an apparent indication. The system, however, *will* generate a critique when a test apparently indicated is not performed.

HyperCritic generates comments in a two-stage process. First, the system interprets the medical record to discover the actions of the GP at a given visit (for example, starting a new drug, continuing treatment with a drug, or replacing one drug with another drug). Second, to review each action, HyperCritic (1) searches the medical record for conditions that contraindicate that action (for example, contraindications to specific drugs), (2) determines whether preparations required for the action have been performed (for example, the evaluation of the kidney function before initiation of treatment with an ACE-inhibitor), (3) determines whether the GP has performed the routine monitoring required by the action (for example, when continuing treatment with thiazides, monitoring the potassium level), and (4) searches for any undesirable condition that might have resulted from the action (for example, the occurrence of a side effect of a drug).

Reviewing the computer-stored medical record requires specific medical knowledge. Consequently, HyperCritic contains detailed drug information (for example, customary dosages, contraindications, side effects, interactions), work-up requirements for hypertensive patients, and criteria for judging the efficacy of the treatment. A detailed description of the system and examples of its output have been presented elsewhere [13].

5.2.3 Patients

The installed base of ELIAS systems includes over 200 sites. Although this is a large number of systems, most ELIAS installations have been established in the past 2 years. When a GP or group of GPs introduces ELIAS into the practice, included in the automated medical record is a summary of each patient's prior medical history; this summary is not representative for the medical record as maintained by the GP in day-to-day use of the system. In the current study, we therefore selected the two oldest ELIAS installations (1985 and 1986) as a source of patient data. The systems are used in two group practices by eight Dutch GPs who provide primary care for approximately 13,000 patients. The practices no longer maintain paper medical records, as the GPs enter all pertinent clinical data directly into the computer at the time of each clinical encounter.

We selected from this population patients who met the following eligibility criteria:

- The patient had an explicit diagnosis of hypertension (that is, the GP stated in the medical record that the patient suffered from hypertension).
- ELIAS was in use when hypertension was first diagnosed. (If the hypertension was diagnosed prior the introduction of ELIAS, data describing workup and initial treatment will be available only in a summarized form.)
- The GP had not referred the patient to a hospital-based physician for evaluation of the hypertension. (The data related to the hospital admission often will be available only in a summarized form.)
- The patient was not a pregnant woman. (HyperCritic does not contain specific knowledge of the treatment of hypertension during pregnancy.)

During the year following the introduction of ELIAS into these clinics (1986 and 1987, respectively), there were 183 patient cases that met these eligibility criteria, from which we selected 20 cases at random. We limited the number of cases to 20 due to restrictions on the time available to the physicians participating in the study. Given the 20 cases, we estimated the total workload for each participating physician at 25 to 35 hours. The study period covered all visits to the treating GPs following the introduction of ELIAS until 1989.

The patients' ages ranged from 21 to 80 years (median 61). The number of

visits to the GP per patient during the study period ranged from 4 to 26 (median 11). Eight patients were female; 12 were male.

5.2.4 Review by Physicians

Printouts of the 20 medical records were submitted to eight physicians. Two of the eight were senior internists with an interest in cardiovascular diseases (witnessed by publications in this area). These two specialists, *physicians I* and *II*, worked in different university hospitals in different parts of The Netherlands. Three of the eight were GPs working in institutes for primary care.[1] These three GP's, *physicians III*, *IV*, and *V*, had an interest in the treatment of cardiovascular diseases in primary care (witnessed by publications in this area); they worked in different practices in different parts of The Netherlands. These five physicians nominated practicing GPs who had no position (full-time or part-time) in any medical school, from whom we selected three at random: *physicians VI*, *VII*, and *VIII*.

We randomized the order in which each physician reviewed the 20 patient records. Figure 1 shows a segment of one of the medical records that was reviewed. The physicians worked independently, indicating on a questionnaire whether each examination or therapeutic action was appropriate or inappropriate, and whether any tests or interventions seemed to be missing from the patient record. When the reviewers deemed an item to be inappropriate or missing, they justified their assessment. For each visit documented in the medical record, they wrote any other comments in free text.

For each visit, we prepared a list of comments based on the reviews of all eight physicians. We considered a comment to be an individual remark regarding a specific action (or the absence of an action) described in the medical record. For example, "I would not treat this patient with this drug; if you insist on treating the patient with this drug, then the dosage is too high," is two comments: (1) Treatment with this drug should not be started, and (2) The dosage of the drug should be reduced.

---

[1]In The Netherlands, general practice is regarded as a medical specialty. To become a GP, a person must complete 2 years of specialty training. The Dutch medical schools have institutes for primary care that are responsible for training these GPs.

We assigned each comment to one of three groups: (1) *diagnosis comments* dealt with the diagnosis of hypertension, (2) *selection comments* dealt with selection of the optimal treatment for the patient, and (3) *execution comments* dealt with the execution of the treatment (for example, the dosage of drug to prescribe, the appropriate precautions, the side effects).

5/29/87	Subjective	*Visit for monitoring blood pressure.*
	Objective	BP 135/100, pulse 76, weight 69.5 kg
	Plan	Capoten® 25 mg TID
8/20/87	Subjective	*Visit for monitoring blood pressure.*
	Objective	BP 160/100, pulse 72
	Plan	Capoten® 50 mg TID, Moduretic® QD
1/20/88	Subjective	*Visit for monitoring blood pressure.*
	Objective	BP 135/85, pulse 72
	Plan	Capoten® 50 mg TID, Moduretic® QD
4/8/88	Subjective	*Visit for monitoring blood pressure.*
	Objective	RR 140/80, pulse 72
	Plan	Capoten® 50 mg TID, Moduretic® QD
5/26/88	Subjective	*Rash on forearms.*
	Assessment	Photosensitivity (3660); *possibly caused by Capoten®.*
	Plan	Continue treatment. *Request lab.*
6/3/88	Subjective	*Still rash on forearms. Not all parts that have been exposed to sunlight have been affected.*
	Objective	Creat 138 micrmol/l, BUN 9.4 mmol/l, ESR 6 mm/hr, HCT 0.45, Hb 8.8 mmol/l, LDH 219 U/l, SGOT 14 U/l, SGPT 9 U/l, Alk phos 39 U/l, Leuco count 7.5 /nl, diff: eos 3, segs 68, lymphs 24, monos 5. Ery count 4.92 /pl. Ery: no abnormalities.
	Assessment	*Photosensitivity less probable. Not all parts effected. Has already used Capoten® for a long period.*
	Plan	Continue treatment.

Figure 1: A segment of an ELIAS medical record. In this segment, not all data in the patient's chart are available in a coded format; the data that are not coded (free text) are shown in *Italics*.

In a second Delphi-type round [18], we submitted to each reviewing physician the same 20 patient cases, with the list of comments for each visit that had been generated by the reviewers during the first round. Each physician then stated whether he believed each comment to be correct.

5.2.5 Review by HyperCritic

We submitted the same 20 patient records to HyperCritic. We mapped the

comments of the computer system onto the comments of the eight reviewing physicians. Three independent referees from different universities verified this mapping. If two of the three referees disagreed with the mapping as proposed by the investigators, then the mapping was adjusted accordingly.

For example, Table 1 shows the comments of the physicians based on the segment of the ELIAS medical record shown in Figure 1. The first column, *Visit Date*, specifies the date of the visit. The second column, *N*, shows how many of the eight physicians agreed with the comment. The third column, *HyperCritic*, shows whether HyperCritic generated the same comment: + indicates that HyperCritic made the same comment. The last column, *Comment*, describes the content of the comment.

We could not map all of HyperCritic's comments to the previous comments of the critiquing physicians. We submitted those comments that we could not map to the eight physicians. We did not inform the physicians that these comments were generated solely by a computer system. We simply told them that "additional comments had been received." Each physician indicated whether or not he believed each comment to be correct.

Table 1: Comments Generated by Physicians

Visit Date	N	Hyper-Critic	Comment
08/20/87	5	+	Do not combine a potassium-sparing diuretic with an ACE-inhibitor.
08/20/87	7	+	Due to the treatment with the diuretic, measure the potassium.
08/20/87	2	-	The blood-pressure level is acceptable. Do not increase the drug regimen.
08/20/87	2	+	The increment in the dose of captopril (from 75 to 150 mg) is too great.
08/20/87	5	-	The dose of captopril is high already (75 mg); increasing the dose to 150 is useless.
04/08/88	8	-	Blood pressure has responded well to treatment. Reduce drug doses.
05/26/88	4	+	Photosensitivity could be caused by captopril; discontinue treatment with captopril.
05/26/88	2	-	Photosensitivity could be an allergic reaction; discontinue all drugs.

5.2.6 Analysis

An ideal assessment would compare the comments against a gold standard. No gold standard for the treatment of hypertension is available, however. In our analysis, we used two different "silver standards" for judging performance. First, we compared the output of HyperCritic with the combined opinion of the physicians to identify the reasons that HyperCritic did not generate certain comments that the majority of the reviewing physicians had agreed were correct. Our approach required us to divide the physicians' comments into three classes. An *accepted comment* was judged correct by six or more of the eight physicians; a *debatable comment* was judged correct by four or five physicians; and a *rejected comment* was judged correct by three or fewer physicians.

For each accepted comment not made by HyperCritic, we performed an analysis to establish why the system had failed to make that comment. When the comment was outside the predefined domain of the system, we labeled the comment *outside domain*. When the comment was inside the domain of the system but required the interpretation of free text, we labeled the comment *free-text dependent*. When the comment was inside the domain and did not require the interpretation of free text, we determined whether the system did consider that comment but, due to a threshold in the system, refrained from making the comment. If so, we labeled that comment *considered*, and we identified the threshold that, if changed, would cause HyperCritic to produce that comment. Finally, when the comment failed to fall into any of the above categories, we labeled it *omission* and we identified the knowledge required by HyperCritic to generate that comment.

The second assessment method that we used to understand differences between the computer system and the peer reviewers compared individual sources of comments (a physician or HyperCritic) with the other sources of comments; we shall refer to each individual source of comments as an *observer*. We removed from the total set of comments those that were *outside domain* and *free-text dependent*. For the remaining comments, we assumed that the failure of HyperCritic to generate a given comment was equivalent to a reviewing physician judging that comment to be incorrect. Under this assumption, we were able to use Kappa statistics to assess the concurrence among all observers (including the computer program). Kappa statistics adjust for chance agreement between observers, and can be used to measure interobserver variation. We calculated the unweighted Kappa value for each pair

of observers [19]. Subsequently, we performed a hierarchical cluster analysis [20]. We then applied Schouten's procedure for identifying homogeneous subgroups of observers [20]; this procedure relies on an unweighted Kappa value when a particular observer is compared to the other ones. A homogeneous subgroup of observers could be obtained by successive removal of observers from the group in a stepwise manner.

Although the Kappa value provides a measure of the degree of agreement among observers, the nature of any disagreement is not clarified. We therefore assumed that the majority opinion of the observers (the HyperCritic program included) correctly determined in each case whether a given comment was appropriate. Under this additional assumption, it was possible to estimate the sensitivity, specificity, and predictive value of each of the observers in generating appropriate comments.

5.3 RESULTS

The physicians made 437 comments, an average of 1.8 comments per patient visit. The number of comments per patient record ranged from 7 to 70 comments (median 19). Of the 437 comments, 138 also were made by HyperCritic. In addition to these 138 comments, HyperCritic produced 31 unique comments (total comments 468).

To provide a flavor of the comments generated by the computer program alone, Table 2 shows the unique comments offered by HyperCritic based on the ELIAS medical record shown in Figure 1. The first column, *Visit Date*, specifies the date of the visit. The second column, *N*, shows how many of the eight physicians agreed with the comment. The last column, *Comment*, describes the content of the comment.

5.3.1 HyperCritic and Majority Opinion

Of the 437 comments made by the physicians, 240 were judged correct by 6 to 8 physicians (*accepted* comments), 108 were judged correct by 4 to 6 physicians (*debatable* comments), and 89 were judged correct by 1 to 3 physicians (*rejected* comments).

Of the 240 *accepted* comments, HyperCritic reproduced 98 (41 percent); of the 108 *debatable* comments, HyperCritic reproduced 34 (31 percent); and of the 89 *rejected* comments, HyperCritic reproduced 6 (7 percent). In addition, HyperCritic produced a total of 31 comments not made by any of the critiquing physicians. Of these 31 comments, 20 were judged correct by 6 to 8 physicians (*accepted* comments); 7 were judged correct by 4 to 5 physicians; and 4 were judged correct by 1 to 3 physicians.

Table 2: Unique Comments Made by HyperCritic.

Visit Date	N	Comment
08/20/87	6	The initial dose of hydrochlorothiazide is too high (1 tablet Moduretic® contains 50 mg hydrochlorothiazide and 5 mg amiloride). The recommended initial dose of hydrochlorothiazide is 12.5 to 25 mg per day.
01/20/88	8	The interval between this visit and the previous visit is too long. You changed the drug regimen on the previous visit. After a change in drug regimen, the effect should be measured within 6 weeks.
05/26/88	5	You are treating this patient with anti-hypertensive drugs. You should measure the blood pressure.
06/03/88	5	You are prescribing captopril. Captopril may cause a decrease in kidney function. The creatinine has increased (138 micromol/l). Consider appropriate action.
06/03/88	5	The creatinine has increased (138 micromol/l). Decreased kidney function is a contraindication to the use of potassium-sparing diuretics. Reconsider your choice.

Of the total of 260 accepted comments, HyperCritic failed to reproduce 142 (Table 3). Of the 139 accepted comments dealing with *diagnosis,* the system failed to reproduce 93. Eighty comments were outside the domain of the system. Five comments required the interpretation of data available only in free text. In six cases, HyperCritic did entertain the possibility of the comment but, due to a specific threshold in the system, refrained from making that comment; these comments dealt with the minimum workup requirements for hypertension. Twice HyperCritic failed to reproduce a comment due to omissions in its knowledge base: In both cases, the system failed to detect a possible primary cause of the patient's hypertension.

Of the 52 accepted comments dealing with the *selection of therapy*, HyperCritic

showed the highest specificity, 0.88, and the highest predictive value of a critique, 0.94. The physicians showed an Index of Merit ranging from 0.30 to 0.56, HyperCritic showed an Index of Merit of 0.62.

Table 5: Identification of a Homogeneous Subgroup of Observers.

Step	Overall	Ph-I*	Ph-IV	Ph-VII	Ph-V	Ph-VIII	Ph-VI	Ph-II	Ph-III	SYST
0	0.20	0.13	0.16	0.19	0.19	0.20	0.20	0.21	0.21	0.25
1	0.21		0.17	0.19	0.22	0.21	0.21	0.21	0.22	0.25
2	0.25			0.19	0.32	0.20	0.32	0.23	0.23	0.27
3	0.23				0.19	0.20	0.21	0.25	0.25	0.31
4	0.25					0.18	0.20	0.28	0.29	0.33
5	0.31						0.17	0.37	0.37	0.39
6	0.45							0.46	0.44	0.45

*Ph-I through Ph-VIII refer to physicians I through VIII; SYST refers to HyperCritic.

Table 6: Comparison of Observers to the Majority Opinion.

	Sensitivity*	Specificity	PV positive	PV negative	Index of Merit
Ph-I[§]	0.94	0.36	0.79	0.70	0.30
Ph-II	0.86	0.70	0.88	0.66	0.56
Ph-III	0.72	0.82	0.91	0.54	0.54
Ph-IV	0.65	0.75	0.87	0.46	0.40
Ph-V	0.73	0.69	0.86	0.50	0.42
Ph-VI	0.70	0.78	0.89	0.50	0.48
Ph-VII	0.88	0.52	0.82	0.63	0.40
Ph-VIII	0.74	0.77	0.89	0.54	0.51
SYST	0.74	0.88	0.94	0.57	0.62

* From the comments judged positive by the majority, we defined the *sensitivity* of an individual observer as the fraction of those comments with a positive judgment by that observer as well. From the comments judged negative by the majority, we defined the *specificity* of an individual observer as the fraction of those comments with a negative judgment by that observer as well. From the comments judged correct by an individual observer, we defined the *predictive value of a positive judgment* (the column labeled *PV positive*) as the fraction of those comments with a positive majority opinion as well. From the comments judged negative by an observer, we defined the *predictive value of a negative judgment* (the column labeled *PV negative*) as the fraction of those comments with a negative majority opinion as well. The *Index of Merit* is defined as (Sensitivity plus Specificity) minus 1; this index thus ranges from -1 to +1.

§ Ph-I through Ph-VIII refer to physicians I through VIII; SYST refers to HyperCritic.

5.4 DISCUSSION

The development of computer-based tools to assist physician decision making has been a focus of intense research for over two decades [2]. Recently, computer programs that aid diagnosis and therapy have become available commercially. Although useful computer programs are beginning to emerge, clinicians are hesitant to bring such medical software into their practices.

The barriers to the widespread adoption of computer-based decision support are substantial [21]. Researchers have made progress in modeling medical knowledge, in representing that knowledge within a computer, and in developing easy-to-use computer-physician interfaces. Workers in artificial intelligence in medicine, however, identify as one of the primary limitations in the spread of decision-support technology the inability of current computer programs to acquire clinically relevant data automatically from hospital and office information systems [22].

The HyperCritic program is the result of an experiment: A decision-support system has been integrated directly into an automated office information system that is in widespread use in The Netherlands. The current study was designed both to investigate whether the data available from the computer-based medical records were sufficient to generate clinically useful comments concerning the primary-care physicians' therapy, and to validate the HyperCritic approach to review of physicians' decision making.

The electronic ELIAS medical records did contain sufficient information for both human observers and HyperCritic to generate substantial critiques. The number of comments indicates that there may be deficiencies in the management of hypertensive patients, and that both human observers and HyperCritic can provide comments that are useful to physicians. Other researchers also have shown that computer-based review is feasible and has a beneficial effect on the delivery of care [10,23-27].

The limitations of computer-based reviewing can be addressed from three perspectives: (1) limitations due to the data in the computer-stored medical record, (2) limitations due to available medical knowledge, and (3) limitations due to the reviewing system itself.

5.4.1 Limitations Due to Data in Medical Record

A principal barrier to the HyperCritic approach is the difficulty the computer system has in determining the goals of the treating physician. Knowledge of those goals is essential in inferring the appropriateness of a physician's actions. Modeling the physician's intentions not only is a problem for HyperCritic, but also presents a problem to peer reviewers.

When people interact with their environment, they form models of themselves and of the environment with which they are interacting. Such internal models are known as *mental models*. Mental models provide predictive and explanatory power for understanding the interaction with the world. Thus, physicians form mental models of the patients whom they treat, and of their own roles in the treatment of their patients [28]. When another physician, or a computer system, is asked to review a physician's treatment, he (or it) has to reconstruct the intentions and reasoning of the treating physician. The reviewer has to formulate a model of the treating physician's mental model. Such a model of a mental model is known as a *conceptual model* [29]. The creation of a conceptual model of the treating physician lies at the heart of a critique: It involves the recognition of the intentions (the treatment objectives) of the physician in combination with the actions undertaken to achieve those objectives [30].

Medical records contain both data describing the patient's state (for example, the results of laboratory tests) and data describing the objectives of the treating physician (for example, a list of treatment goals). The medical record, however, is primarily a record of the actions performed; it records "what I did," rather than "why I did it." A physician often does not label a particular prescription "for the treatment of hypertension," and does not necessarily label diagnostic tests "to exclude disease X." Moreover, a physician does not record in the medical record all actions taken or all decisions made. One of the goals of the problem-oriented medical record [31] is to facilitate mutual understanding among different physicians by introducing the notion of a *problem* as one of the axes along which to structure the medical data. By assigning the entries in the medical record to the problems that the physician has identified, health practitioners can clarify some of the ambiguity of those entities. If a patient has multiple diseases and a given therapeutic intervention might be appropriate for more than one of those conditions, then a problem-oriented medical record requires the physician to ascribe the therapeutic intervention explicitly to the particular disease for which it is intended. When a physician prescribes a beta-

blocker for a hypertensive patient who has situational anxiety, the problem-oriented approach would enable a reviewer to identify whether that drug was targeted at the hypertension or at the anxiety (or perhaps both).

The designers of current medical-record systems, however, have not anticipated that a computer system might need to construct a conceptual model of the treating physician based on the medical-record data. That a computer system might need to issue claims for reimbursement usually has been anticipated, however. All data required to generate insurance claims are thus available in a coded format (for example, type of visit, type of insurance, procedures performed). Typically, the data in the computer-based medical record that describe the reasoning of the physician are in the form of free text and cannot be linked to other data in the medical record. Because the designers assumed that only humans would attempt to reconstruct the reasoning of the physician, they placed little emphasis on how that reasoning could be captured in a structured and coded -- and, consequently, machine-interpretable -- form. (The PROMIS system developed by Weed [32] is a notable exception.)

The absence of coded data in the medical record that can allow the computer to create a comprehensive conceptual model of the treating physician represents a fundamental obstacle to computer-based critiquing; it caused the designers of HyperCritic to *exclude certain comments* from the domain of the system. Thus, HyperCritic will never judge a diagnostic investigation to be inappropriate based solely on the absence of an apparent indication. To produce such a statement, the critiquing system would have to perform an overall analysis of the patient to construct a detailed conceptual model of the treating physician's thoughts. HyperCritic, with knowledge limited to the domain of hypertension management and access to only coded data in the computer-based medical record, is unable to perform such an overall assessment; HyperCritic reviews only bits and pieces of the care that the patient is receiving (for example, verifying a minimum workup, searching for side effects of drugs, searching for contraindications to therapy, or interpreting laboratory results).

Although physicians are able to interpret free text in the automated medical record, HyperCritic is not able to interpret uncoded data. In the current ELIAS medical record, a substantial portion of the patient-history data is available only in free text. Consequently, the ability of HyperCritic, as compared to physicians, to comment on the diagnostic workup and on the selection of drugs is restricted. ELIAS does not yet contain a coding scheme for the nondrug treat-

ment of hypertension. Consequently, HyperCritic cannot provide any comments dealing with this important aspect of patient care. Moreover, HyperCritic cannot comment on the decision of the physician to change from nondrug to drug treatment of hypertension.

5.4.2 Limitations Due to Available Medical Knowledge

In addition to the restrictions due to the medical record, computer-based critiquing is also limited by available medical knowledge. The low Kappa values between different reviewers (see Table 4) underline the observation that many decisions made by physicians appear arbitrarily variable, and that this arbitrariness represents, for at least some patients, suboptimal or even harmful care [33]. The problem, of course, is that often there are no data that allow us to identify the subsets of patients that receive less-than-optimal management. In many areas of medicine, there simply is no consensus that defines proper therapy. Yet critiquing requires a set of criteria by which to critique. The development of a critiquing system forces the explicit definition of those criteria. McDonald reports that computer-based review has had significant effect on care only when physicians agreed a priori about the ideal approach to a problem [34,35].

The absence of a gold standard for many aspects of physician decision making complicates the evaluation of automated decision aides such as HyperCritic. For example, developers of the MYCIN computer system for diagnosis and treatment of meningitis found that, although a panel of eight infectious-disease specialists viewed the computer's suggested treatment of meningitis to be acceptable in 65 percent of cases, the same panel viewed the treatment suggested by their colleagues on the panel to be *acceptable* in only 43 to 63 percent of cases [3].

In the absence of an absolute standard of care, the developers of a critiquing system -- like the developers of paper-based practice guidelines -- are confronted with a range of opinions that reflect personal preferences, practice variation, and subjective evaluation [34]. HyperCritic embodies just one set of opinions in a range of possible choices. The system thus contains one set of criteria for reaching particular conclusions, whereas some critiquing physicians apparently use other sets of criteria. In the current study, HyperCritic tended to be more lenient (that is, less critical) than were the critiquing physicians. The

relatively low sensitivity achieved by HyperCritic (see Table 6) illustrates this forgiving nature of the system.

HyperCritic provides decision support as a byproduct of routine data-management activities. The physician does not specifically ask for advice. Therefore, a goal and challenge is to avoid generating excessive numbers of comments -- particularly because false-positive advisory reports may generate antagonistic responses. Consequently, for systems such as HyperCritic, ensuring that critiques have high predictive value is of great importance. The dilemma, of course, is that increasing the sensitivity of the computer system will, in all likelihood, decrease its predictive value by causing the generation of increased numbers of false-positive comments.

5.4.3 Limitations Due to HyperCritic

We have identified several areas in which HyperCritic failed due to omissions or incompleteness in the system's knowledge base. Underlying HyperCritic is a model of the treatment of hypertension. Models are, by their nature, limited in the entities that they contain. System builders may decide not to model certain entities, and to inform the user of those limitations [36]. (For example, HyperCritic will not comment on the basis for the physician's diagnosis of hypertension.) Identification of the entities that the system designers omitted inadvertently, however, requires evaluation of the computer system. We discovered several such inadvertent omissions during this evaluation of HyperCritic.

HyperCritic requires a better model for searching for possible primary causes of hypertension, and for monitoring of a patient after the physician stops antihypertensive drugs. The current version of HyperCritic contains only a specification of the diagnostic investigations required by a minimum workup. Consequently, if all findings required by a minimum workup are normal, the system must assume that the treating physician's diagnosis of essential hypertension is correct. In the current study, one patient had just sustained a severe blow to the head resulting in a cerebral contusion. This condition may cause a period of high systolic blood-pressure readings. HyperCritic failed to identify the history of trauma as a possible cause of the transient rise in blood pressure. The fundamental issue is that HyperCritic contains a model of the typical workup for hypertension, rather than a model of all possible causes of

elevated blood-pressure readings.

That HyperCritic does not include any medical knowledge dealing with how to monitor a patient once drug treatment has been discontinued is an error on the part of the designers of the system. We inadvertently omitted such knowledge.

5.5 CONCLUSIONS

The ELIAS medical records did contain sufficient information for human observers and for the HyperCritic computer program to generate substantial critiques of hypertension management. Comparing the performance of HyperCritic with the performance of physicians shows that HyperCritic, in its limited domain, can compete successfully with human observers. HyperCritic, however, is unable to assess the overall condition of the patient; HyperCritic reviews only fragments of the care that the patient is receiving.

Further development of computer-based critiquing requires that two main tasks be addressed: the development of a computer-based medical record that captures the reasoning pattern of the physician, and the development of widely accepted practice guidelines. In the current computer-based medical record, data describing the thought patterns of the physician often are scanty and, if available, usually are in the form of free text. Only when the reasoning process of the physician is available in a formal notation will an automated critiquing system be able to perform its review in the context of the intentions of the attending physician. Critiquing, however, is possible only in the presence of accepted practice guidelines. In the absence of such guidelines, critiquing is limited to expressing just another personal opinion.

Pressure form law-enforcing bodies, from third-party payers, from peer-review organizations, from hospitals, and from physicians and patients themselves may lead to the increasing use of automated review of medical records. As such technology is put into place [1,37], we must remember that computer systems are based on models, and that models are limited in the entities that they contain. There is no evidence that the capabilities of machines will ever approach those of humans in dealing with unexpected situations, in understanding the patient in his social context, in integrating the often complex and confounding presentation of a disease into a coherent pattern, or in dealing with ethical issues [38]. A priori, system developers may choose not to model

certain aspects of medical care, and to inform their users of these limitations in their systems. We still will need to evaluate such systems thoroughly, however, to identify the medical knowledge that the system builders have omitted inadvertently. Before any computer-based decision-support system is released, a formal evaluation using real patient data must be performed so that the limitations of that system will be discovered [39].

ACKNOWLEDGMENTS

We are indebted to those GPs who made their medical records available for scrutiny by other physicians, and we acknowledge the efforts of the physicians who participated in this study.
We thank B.M.T. Mosseveld for his competent programming assistance, and Ms R.J.J.R. Scholte for secretarial support.
Support of the Netherlands Heart Foundation (Grant 88.236) is gratefully acknowledged.

REFERENCES

[1] McDonald CJ, Tierney WM. Computer-stored medical records: Their future role in medical practice. *JAMA* 1988;259:3433-40.

[2] Shortliffe EH. Computer programs to support clinical decision making. *JAMA* 1987;258:61-6.

[3] Yu VL, Fagan LM, Wraith SM, et al. Antimicrobial selection by a computer: A blinded evaluation by infectious disease experts. *JAMA* 1979;242:1279-82.

[4] Miller RA, Pople HE Jr, Myers JD. Internist-1, An experimental computer-based diagnostic consultant for general internal medicine. *N Engl J Med* 1982;307:468-76.

[5] Miller RA, Masarie FE. The demise of the Greek oracle model for medical diagnosis systems. *Meth Inform Med* 1990;29;1-2.

[6] Clancey WJ, Shortliffe EH. *Readings in Medical Artificial Intelligence: The First Decade.* Reading, MA: Addison-Wesley, 1984.

[7] Szolovits P. *Artificial Intelligence in Medicine.* AAAS Selected Symposia Series. Boulder, CO: Westview Press, 1982.

[8] Miller PL. *Expert Critiquing Systems: Practice-Based Medical Consultation by computer.* New York: Springer-Verlag, 1986.

[9] Clyman JI, Black HR, Miller PL. Assessing practice conformance for hypertension management using an expert system. In: *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care.* Washington DC: IEEE Computer Society Press, 1989:124-8.

[10] Evans RS, Larsen RA, Burke JP, et al. Computer surveillance of hospital-acquired infections and antibiotic use. *JAMA* 1986;256:1007-11.

[11] Musen MA. The strained quality of medical data. *Meth Inform Med* 1989;29:123-5.

[12] Burnum JF. The misinformation era: The fall of the medical record. *Ann Int Med* 1989;110:482-4.

[13] Van der Lei J, Musen MA. A model for critiquing based on automated medical records. Rotterdam, The Netherlands: Erasmus University, Department of Medical Informatics, Technical Report MIEUR-90-6. 42 pp. *Submitted for publication.*

[14] Westerhof HP, Boon WM, Cromme PVM, et al. ELIAS: Support of the Dutch general practitioner. In: Reichertz PL, Engelbrecht R, Picollo U, eds. *Present Status of Computer Support in Ambulatory Care.* New York: Springer-Verlag, 1987:1-10.

[15] Duisterhout JS, Branger PJ. Communication in primary care systems. In: Barber B, Cao D, Qin D, Wagner G, eds. *MEDINFO 89.* Amsterdam: North-Holland, 1989:700-3.

[16] Lamberts H, Wood M, eds. *International Classification of Primary Care.* Oxford, England: Oxford University Press, 1987.

[17] Royal College of General Practitioners. Classification of diseases, problems and procedures. London, England: Royal College of General Practitioners, Occasional paper 26, 1984.

[18] Dalkey NC. An experimental study of group opinion: The Delphi method. *Futures* 1969:408-26.

[19] Cohen J. A coefficient of agreement for nominal scales. *Educ Phychol Meas* 1960;20:37-46.

[20] Schouten HJA. Measuring pairwise interobserver agreement when all subjects are judged by the same observers. *Statistica Neerlandica* 1982;36:45-61

[21] Shortliffe EH. Testing reality: The introduction of decision-support technology for physicians. *Meth Inform Med* 1989;28:1-5.

[22] Cooper GF, Musen MA. The 1990 AAAI spring symposium on artificial intelligence in medicine. *Artif Intel Med* 1990;2;293-8.

[23] McDonald CJ, Hui SL, Smith DM, et al. Reminders to physicians from an introspective computer medical record. *Ann Intern Med* 1984;100:130-8.

[24] Barnett GO, Winickoff R, Dorsey JL, Morgan MM, Lurie RS. Quality assurance through automated monitoring and concurrent feedback using a computer-based medical information system. *Med Care* 1978;16:962-70.

[25] McDonald CJ. Protocol-based computer reminders: the quality of care and the non-perfectibility of man. *N Engl J Med* 1976;295:1351-4.

[26] Pryor TA, Gardner RM, Clayton PD, et al. The HELP system. *J Med Sys* 1983;7:87-102.

[27] Warner HR. *Computer-Assisted Medical Decision Making.* New York: Academic Press, 1978.

[28] Norman DA. Some observations on mental models. In: Gertner D, Stevens AL, eds. *Mental Models.* Hillsdale, NJ: Lawrence Erlbaum, 1983:7-14.

[29] Gertner D, Stevens AL, eds. *Mental Models.* Hillsdale, NJ: Lawrence Erlbaum, 1983.

[30] Miller PL. Goal-directed critiquing by computer: ventilator management. *Comp Biomed Res* 1985;18:422-38.

[31] Weed LL. *Medical Records, Medical Education, and Patient: The Problem-Oriented Record as a Basic Tool.* Cleveland, OH: Case Western Reserve Press, 1971.

[32] Weed LL. Representation of Medical Knowledge in PROMIS. In: Blum BI, ed. *Computers and medicine: Information Systems for Patient Care.* New York: Springer Verlag, 1984:83-108.

[33] Eddy DM. Clinical decision making: From theory to practice. The challenge. *JAMA* 1990;263:287-90.

[34] McDonald CJ. *Action-Oriented Decisions in Ambulatory Medicine.* Chicago: Year Book Medical Publishers, 1981.

[35] McDonald CJ, Wilson GA, McGabe GP. Physician response to computer reminders. *JAMA* 1980;244:1579-81.

[36] Musen MA, Van der Lei J. Knowledge engineering for clinical consultation programs: Modeling the application area. *Meth Inform Med* 1989;28:28-35.

[37] Barnett GO. The application of computer-based medical-record systems in ambulatory practice. *New Engl J Med* 1984;310:1643-50.

[38 Blois MS. Clinical judgment and computers. *New Engl J Med* 1980;303:192-7.

[39] Bemmel JH van, Willems JL. Satndardization and validation of medical decision-support systems: the CSE project. *Meth Inform Med* 1990;29:261-2.

CHAPTER 6

Summary

6.1 SUMMARY

The purpose of this study was the creation of a model for critiquing based on data obtained from computer-stored medical records. The underlying assumption is that data obtained from automated medical records can be used to generate a medically relevant critique. To validate our ideas, we developed a system, HyperCritic, that critiques the decision making of general practitioners (GPs) caring for patients with hypertension.

Developers of computer-based decision-support tools frequently adopt either pattern recognition or artificial-intelligence techniques as the basis for their programs. Because these developers often choose to accentuate the differences between these alternative approaches, the more fundamental similarities are frequently overlooked. The principal challenge in the creation of any clinical consultation program -- regardless of the methodology that is used -- lies in creating a computational model of the application domain. The difficulty in generating such a model manifests itself in symptoms that workers in the expert-systems community have labeled "the knowledge-acquisition bottleneck" and "the problem of brittleness." In **Chapter 2**, we explore these two symptoms, and show how the development of consultation programs based on pattern-recognition techniques is subject to analogous difficulties.

In **Chapter 3**, we report a study in which a general practitioner (GP) was asked to provide us with the computer-based medical records of five patients with hypertension. A printout of these medical records was submitted to an internist who had a recognized interest and experience in the treatment of hypertension. The internist was asked to comment on the treatment of the hypertension as documented in the medical records. Subsequently, the comments of the internist were submitted to a panel of three GPs; these GPs were asked to judge the relevance of the comments. Finally the comments of the internist were shared with the GP who had treated the patient.

The internist generated 48 comments. When the GPs were asked to judge the comments of the internist, over 50% of these comments were judged relevant - but there was little consensus among the GPs regarding which comments were the relevant ones. The GPs were also asked to state why a given comment was not relevant. Over 90% of their reasons fell into the following three groups: (a) the GP disagreed with the advice, (b) the GP agreed with the principle but he would prefer to modify the recommendation to suit his practice setting, or (c)

the GP felt that the advice had no consequence for the decision he had to make, although he did not disagree with the underlying principle. The treating physician judged over 50% of the internist's comments relevant. The predominant reason why the treating physician judged a comment to be irrelevant or less relevant was a misunderstanding of his intentions and/or reasoning by the critiquing physician.

In **Chapter 4**, a model for critiquing based on computer-stored medical records is presented. The core of our model is that two distinct types of knowledge are needed during the critiquing process: knowledge about the process of critiquing itself (that is, *critiquing knowledge*) and specific medical knowledge that will be required during the critiquing process (that is, *medical knowledge*). Critiquing knowledge describes when and how to critique, whereas medical knowledge provides the factual knowledge required during the critiquing process. Critiquing knowledge describes the *process* of critiquing; medical knowledge describes the *content* of critiquing.

We identified four types of domain-independent critiquing tasks: preparation tasks, selection tasks, monitoring tasks, and responding tasks. These tasks can be distinguished from the medical knowledge that a critiquing program uses to generate comments concerning a physician's therapy. These critiquing tasks are designated to detect conflicts between the inferred condition of the patient and the recorded decisions the physician has made. The structure of these critiquing tasks can be separated from the actual medical knowledge required to execute those tasks.

The use of preparation tasks hinges on the notion that certain actions or decisions of the physician should be preceded by one or more preparations or observations. Selection tasks embody the notion that there are situations in which certain actions of the physician may not be the most appropriate. Monitoring tasks represent the notion that a given action of a physician may require subsequent actions (for example, observing a particular patient parameter) at particular intervals. Responding tasks incorporate the notion that some finding related to a physician action may require a response. Note the difference between monitoring tasks and responding tasks: Monitoring tasks report the absence of a particular patient parameter and recommend observation of that parameter, whereas responding tasks report an abnormal or changed value of a parameter and recommend that the physician respond to correct the abnormal value.

attempt to reconstruct the reasoning of the physician, they placed little emphasis on how that reasoning could be captured in a structured, coded (and, consequently, machine-interpretable) form.

The absence of coded data in the medical record that would allow the computer to create a conceptual model of the treating physician represents a fundamental limitation to computer-based critiquing; it caused the designers of HyperCritic to exclude certain types of comments from the domain of the system. HyperCritic will never judge a diagnostic investigation inappropriate solely based on the absence of an apparent indication. To produce such a statement, the critiquing system would have to perform an overall analysis of the patient to construct a comprehensive conceptual model of the treating physician's thoughts. HyperCritic, with knowledge limited to the domain of hypertension and access to only coded data in the computer-based medical record, is unable to perform such an overall assessment; HyperCritic only reviews bits and pieces of the care that the patient is receiving (for example, verifying a minimum workup, searching for side effects of drugs, searching for contraindications to therapy, or interpreting laboratory results).

In addition to the restrictions due to the medical record, computer-based critiquing is also limited by available medical knowledge. The absence of a gold standard for many aspects of physician decision making complicates the development and evaluation of automated decision aides. In the absence of an absolute standard of care the developers of a critiquing system, like the developers of paper-based practice guidelines, are confronted with a range of opinions that reflect personal preferences, practice variation, and subjective evaluation. HyperCritic embodies just one opinion in a range of possible opinions: The system contains one set of criteria for reaching a particular conclusion, whereas some critiquing physicians may use another set of criteria.

HyperCritic provides decision support as a byproduct of routine data-management activities. The physician does not specifically ask for advice. Therefore, a goal and challenge is to avoid generating excessive numbers of comments -- particularly because false-positive advisory reports may generate antagonistic responses. Consequently, for systems such as HyperCritic, ensuring a high predictive value of a critique is of great importance. The dilemma, of course, is that increasing the sensitivity will, in all likelihood, decrease the predictive value of the critique.
We also identified areas in which HyperCritic failed due to omissions or

incompleteness of the system's knowledge base. The current version of HyperCritic contains only a specification of the diagnostic investigations required by a minimum workup. Consequently, if all findings required by a minimum workup are normal, the system assumes that the treating physician's diagnosis of essential hypertension is correct. The fundamental omission is that HyperCritic contains a model of a typical workup rather than a model of possible causes of hypertension.

In **Chapter 5**, we compared each individual source of comments (a physician or HyperCritic) with the other sources of comments. Of the 468 comments, we excluded from further analysis those comments outside the domain of the system and those comments that required the interpretation of free text, leaving 298 comments. In this limited domain, the highest agreement (as measured by the Kappa index) was between two of the eight physicians and HyperCritic. As the Kappa statistics do not provide insight in the nature of the disagreements between the observers, we calculated the sensitivity, specificity, and predictive values of the different sources of critique. HyperCritic shows the highest specificity (0.88) and the highest predictive value of a critique (0.94) when compared to the physicians. The Index of Merit of HyperCritic was higher (0.62) than that of the physicians (ranging from 0.30 to 0.56).

We conclude that the computer-based medical records did contain sufficient information for both human observers and HyperCritic to generate substantial critiques. The frequency of comments indicates that there may be deficiencies in the management of hypertensive patients, and that both human observers and HyperCritic can provide comments useful to physicians. Comparing the performance of HyperCritic with that of physicians shows that HyperCritic, within its limited domain, can successfully compete with human observers.

6.2 CONCLUDING REMARKS

- The core of our critiquing model is the separation of critiquing knowledge from factual medical knowledge. The HyperCritic program is an implementation of that model.

- The additional levels of abstraction introduced to separate critiquing knowledge and medical knowledge can be used for several purposes: to drive knowledge acquisition, to support maintenance of the system, and to

reuse the knowledge.

- Automated medical records did contain sufficient information for human observers and HyperCritic to generate substantial critiques.

- Comparing the performance of HyperCritic with the performance of physicians shows that HyperCritic, within its limited domain, can successfully compete with human observers.

- HyperCritic is unable to assess the overall condition of the patient; HyperCritic only reviews fragments of the care that the patient is receiving.

- Only when the reasoning process of the physician is available in a formal notation will a critiquing system be able to perform its review in the context of the intentions of the physician.

- Critiquing is possible only in the presence of accepted practice guidelines. In the absence of such accepted guidelines, critiquing is limited to expressing just another personal opinion.

- Further development of computer-based critiquing requires two main tasks to be addressed: the development of a computer-based medical record that captures the reasoning pattern of the physician, and the development of widely accepted practice guidelines.

CHAPTER 7

Samenvatting

7.1 SAMENVATTING

Het doel van deze studie was het creëren van een model voor het leveren van kritiek op basis van gegevens uit geautomatiseerde medische dossiers. De onderliggende aanname is dat gegevens uit geautomatiseerde medische dossiers geschikt zijn om medisch relevante kritiek te genereren. Om onze ideeën te toetsen, ontwikkelden we het computer programma HyperCritic. HyperCritic levert kritiek op de beslissingen van de huisarts die patiënten met hypertensie behandelt.

Ontwikkelaars van beslissingsondersteunende systemen gebruiken veelal technieken ontleend of aan de patroonherkenning of aan de kunstmatige intelligentie. Omdat deze ontwikkelaars veelal de verschillen tussen deze technieken benadrukken, worden de fundamentele overeenkomsten vaak over het hoofd gezien. De belangrijkste uitdaging tijdens het creëren van een medisch consultatieprogramma -- onafhankelijk van de techniek die gebruikt wordt -- is het creëren van een formeel model van het toepassingsgebied. De problemen die het gevolg zijn van de moeilijkheid om een formeel model te ontwikkelen manifesteren zich in symptomen die onderzoekers op het gebied van de expert systemen "de kennis-acquisitie flessehals" en "het probleem van de breekbaarheid" hebben genoemd. Na een Inleiding op de studie in **Hoofdstuk 1**, analyseren wij in **Hoofdstuk 2** deze twee symptomen en laten zien hoe consultatieve systemen gebaseerd op patroonherkenningstechnieken analoge problemen onder ogen moeten zien.

In **Hoofdstuk 3** rapporteren we over een onderzoek waarin een huisarts gevraagd werd om vijf geautomatiseerde dossiers uit zijn patiëntenbestand te kiezen, van patiënten met hypertensie. Deze dossiers werden vervolgens aan een internist gegeven van wie bekend was dat hij een uitgesproken belangstelling had voor de behandeling van hypertensie in de eerstelijns gezondheidszorg. De internist werd gevraagd kritiek te leveren op de behandeling zoals gedocumenteerd in het medische dossier. Vervolgens werden de opmerkingen van de internist voorgelegd aan drie huisartsen; deze huisartsen werd gevraagd de relevantie van de opmerkingen te beoordelen. Tenslotte werden de opmerkingen van de internist voorgelegd aan de huisarts die de patiënt behandeld had.

De internist maakte in totaal 48 opmerkingen over de behandeling van de vijf patiënten. Toen de huisartsen gevraagd werd de opmerkingen van de internist

te beoordelen, werd meer dan 50% van de opmerkingen als relevant beoordeeld, maar er was weinig consensus tussen de huisartsen over de vraag, welke opmerkingen de relevante waren. De huisartsen werd ook gevraagd aan te geven waarom een bepaalde opmerking minder relevant was. Meer den 90% van de door hen genoemde redenen vielen in één van de volgende drie categorieën: (a) de huisarts was het inhoudelijk oneens met de opmerking, (b) de huisarts was het in principe eens met de opmerking, maar wilde het concrete advies aanpassen aan de omstandigheden in zijn praktijk, en (c) de huisarts vond dat het advies geen consequenties had voor de beslissingen die hij moest nemen. De huisarts die de patiënten behandeld had beoordeelde meer dan 50% van de opmerkingen van de internist als relevant. De belangrijkste reden die hij aangaf voor het als niet relevant beoordelen van een bepaalde opmerking was dat de internist zijn redeneerproces en bedoelingen niet goed had begrepen.

In **Hoofdstuk 4** beschrijven wij een model voor het leveren van kritiek op basis van gegevens uit geautomatiseerde dossiers. Het hart van ons model is dat twee verschillende soorten kennis nodig zijn tijdens het leveren van kritiek: de kennis omtrent het proces van het leveren van kritiek *(kritiek-kennis)* en de specifieke medische kennis die nodig is tijdens het leveren van kritiek *(medische kennis)*. Kritiek-kennis beschrijft wanneer en hoe kritiek geleverd moet worden; medische kennis beschrijft de feitelijke kennis die noodzakelijk is tijdens het *proces* van bekritiseren. Kritiek-kennis beschrijft derhalve het proces van bekritiseren, medische kennis beschrijft de *inhoud* van de kritiek.

Wij identificeren vier types domein-onafhankelijke kritiek-taken: voorbereidingstaken, selectietaken, bewakingstaken, en reageertaken. Deze taken kunnen gescheiden worden van de medische kennis die het kritiek-programma nodig heeft om opmerkingen over de therapie te maken. Deze kritiek-taken detecteren conflicten tussen de conditie van de patiënt en de gedocumenteerde beslissingen van de arts. De structuur van deze kritiek-taken kan worden gescheiden van de medische kennis die nodig is om de taken uit te voeren.

Voorbereidingstaken maken gebruik van het feit dat bepaalde acties van de arts voorafgegaan moeten worden door een of meer voorbereidingen of observaties. Selectie-taken maken gebruik van het feit dat er situaties kunnen zijn waarin een bepaalde actie van de arts niet de meest geëigende is. Bewakingstaken representeren het feit dat een gegeven actie van de arts vervolg-acties

mentale modellen van zichzelf en de omgeving waarmee ze die interactie aangaan. Dergelijke mentale modellen leveren voorspellende en verklarende inzichten, noodzakelijk voor het begrijpen van de interactie met de omgeving. Ook artsen vormen mentale modellen van de patiënten die ze behandelen, en van hun eigen rol in de behandeling van de patiënt. Wanneer een andere arts (of een computersysteem) gevraagd wordt om de behandeling zoals verricht door hun collega van kritiek te voorzien, dan moet hij of zij (of het) de bedoelingen en het redeneerproces van de behandeld arts a.h.w. reconstrueren. Zo'n model van een mentaal model staat bekend als een *conceptueel model*.

Echter, de ontwikkelaars van geautomatiseerde medische dossiers hebben er niet op geanticipeerd dat een computersysteem een conceptueel model van de behandelende arts zou moeten construeren op basis van de gegevens in het geautomatiseerde medische dossier. Omdat ontwikkelaars aannamen dat alleen mensen zouden pogen het redeneerproces van de arts te reconstrueren, besteedden zij geen aandacht aan de wijze waarop het redeneerproces van de arts geregistreerd zou kunnen worden op een gestructureerde, gecodeerde (en, bijgevolg, door een computer te interpreteren) wijze.

De afwezigheid van gecodeerde data in het medisch dossier die een computerprogramma in staat stellen om een conceptueel model te creëren van de behandelende arts, representeert een fundamentele beperking voor het leveren van kritiek door computersystemen; dit was de reden dat de ontwikkelaars van HyperCritic a priori bepaalde categorieën opmerkingen buiten het domein van het systeem te plaatsten. HyperCritic zal nooit een diagnostisch onderzoek als overbodig rapporteren enkel en alleen omdat er geen indicatie voor dat onderzoek kan worden gevonden in het dossier. Om zo'n opmerking te kunnen maken zou het systeem een totale analyse van de patiënt moeten kunnen maken om een alles omvattend conceptueel model te maken van de moverende redenen van de behandelende arts. HyperCritic, met kennis beperkt tot het gebied van hypertensie en met toegang uitsluitend tot de gecodeerde informatie in het medisch dossier, is niet in staat een dergelijke globale analyse te verrichten; HyperCritic analyseert derhalve slechts fragmenten van de behandeling die de arts voorstelt (bijvoorbeeld, het nakijken van een minimale diagnostiek voordat de behandeling gestart wordt, het zoeken naar bijwerkingen van geneesmiddelen, het zoeken naar contra-indicaties van medicijnen, of het interpreteren van laboratoriumuitslagen).

Naast de beperkingen die veroorzaakt worden door het medisch dossier, is het

leveren van kritiek door een computersysteem ook beperkt door de beschikbare medische kennis. De afwezigheid van een gouden standaard voor vele aspecten van de medische besluitvorming compliceert de ontwikkeling en de evaluatie van beslissingsondersteunende systemen. In de afwezigheid van absolute standaards voor zorg worden de ontwikkelaars van kritiek-systemen, net als de ontwikkelaars van op papier verwoorde behandelprotocollen, geconfronteerd met een breed scala aan meningen die gebaseerd zijn op persoonlijke voorkeur, verschillende praktijken, en subjectieve beoordelingen. HyperCritic representeert slechts één mening temidden van een scala van mogelijke meningen: het systeem bevat een set criteria om een bepaalde conclusie te bereiken, terwijl sommige artsen een andere set criteria hanteren.

HyperCritic geeft ondersteuning als nevenprodukt van een routinematig gebruikte gegevensregistratie. De arts vraagt niet expliciet om kritiek. Een doel en uitdaging is, derhalve, het verhinderen van een te groot aantal opmerkingen -- vooral omdat fout-positieve opmerkingen antagonistische reacties kunnen veroorzaken. Bijgevolg is het voor een systeem als HyperCritic van het grootste belang te zorgen voor een hoge voorspellende waarde van een opmerking. Het dilemma is, uiteraard, dat het verhogen van de sensitiviteit in de regel gepaard gaat met een verlaging van de voorspellende waarde.

We identificeerden ook gebieden waar HyperCritic faalde omdat er tekortkomingen waren in het kennisbestand van het systeem. De huidige versie van HyperCritic bevat alleen een specificatie van de minimaal noodzakelijke diagnostiek voordat de diagnose "essentiële hypertensie" gesteld kan worden. Bijgevolg, als het onderzoek dat bij een dergelijke minimale diagnostiek nodig is normaal is, dan concludeert het systeem dat de diagnose "essentiële hypertensie" terecht is. De fundamentele omissie is dat, alhoewel HyperCritic een model van de minimale diagnostiek kent, het geen model van de mogelijke oorzaken van hypertensie kent.

In **Hoofdstuk 5** vergelijken we ook elke afzonderlijke bron van opmerkingen (een arts of HyperCritic) met de andere bronnen van opmerkingen. Om een dergelijke vergelijking mogelijk te maken, verwijderden wij uit de oorspronkelijke verzameling van 468 opmerkingen die opmerkingen die buiten het domein van het systeem lagen en die opmerkingen die gebaseerd waren op vrije tekst in het medisch dossier; de resulterende verzameling bevatte 298 opmerkingen. In dit beperkte gebied werd de hoogste overeenstemming (gemeten aan de hand van de Kappa index) gevonden tussen twee van de artsen en HyperCritic.

Omdat de Kappa index geen inzicht geeft in de aard van de meningsverschillen, berekenden we ook de sensitiviteit en specificiteit van de verschillende bronnen van kritiek. HyperCritic laat de hoogste specificiteit zien (0.88) en de hoogste voorspellende waarde van een kritiek (0.94). De Index of Merit van HyperCritic was hoger (0.62) dan die van de artsen (variërend tussen 0.30 en 0.56). Ten overvloede herhalen wij dat dit geldt voor het beperkte domein van HyperCritic.

Wij concluderen dat geautomatiseerde medische dossiers voldoende informatie bevatten om substantiële kritiek te kunnen leveren. De frequentie van opmerkingen geeft aan dat er mogelijk deficiënties zijn in de behandeling van hypertensiepatiënten, en dat zowel artsen als HyperCritic in staat zijn relevante opmerkingen te geven. Vergelijking van de prestaties van HyperCritic met die van artsen laat zien dat, in een beperkt domein, het systeem zich kan meten met de artsen.

7.2 AFSLUITENDE OPMERKINGEN

- Het hart van ons kritiek model is de scheiding tussen kritiek kennis en medische kennis. HyperCritic is een implementatie van dat model.

- De additionele abstractieniveaus die worden geïntroduceerd door de scheiding tussen kritiek-kennis en medische kennis kunnen voor diverse doeleinden worden gebruikt: voor het sturen van de kennis-acquisitie, voor het ondersteunen van het onderhoud van het systeem, en voor het vergemakkelijken van het hergebruik van de kennis in het systeem.

- De gebruikte geautomatiseerde medische dossiers bevatten voldoende informatie voor het leveren van substantiële kritiek.

- De vergelijking van de prestaties van HyperCritic met de prestaties van artsen laat zien dat, in een beperkt domein, het systeem zich kan meten met de artsen.

- HyperCritic is niet in staat de algehele conditie van de patiënt te evalueren; HyperCritic analyseert slechts fragmenten van de behandeling die de arts voorschrijft.

- Alleen wanneer het redeneerproces van de arts in een formele notatie

beschikbaar is kan een kritiek-systeem een analyse in de context van de overwegingen van de arts uitvoeren.

- Het leveren van kritiek is alleen mogelijk wanneer er geaccepteerde richtlijnen voor het medisch handelen zijn. Bij afwezigheid van zulke richtlijnen is het leveren van kritiek beperkt tot het leveren van een persoonlijke opinie.

- Verdere ontwikkeling van kritiek systemen vereist dat twee belangrijke problemen dienen worden uitgewerkt: het ontwikkelen van geautomatiseerde dossiers die ook het redeneerproces expliciet vastleggen, en de ontwikkeling van richtlijnen voor het medisch handelen die algemeen geaccepteerd worden.

ABOUT THE AUTHOR

Johannes van der Lei was born in Sneek, the Netherlands, on the 13th of April 1955.

He received his undergraduate education at the Bogerman Lyceum in Sneek (Hogere Burgerschool, B division). He attended medical school at the Free University in Amsterdam ("artsexamen" 1982).

He subsequently held a joint appointment with the departments of Pediatrics and Medical Informatics (1983-1985) developing an information system for the neonatal intensive care unit.

From 1985 to 1988 he worked with the department of Medical Informatics of the Free University in Amsterdam. In this period he started the research reported in this thesis. He moved in 1988 from the Free University in Amsterdam to the Erasmus University in Rotterdam and became member of the newly formed Department of Medical Informatics at that university. At the Erasmus University he completed the research described in this thesis.

Johan van der Lei continues research as scientific staff member of the Department of Medical Informatics at the Erasmus University.