AI Magazine Volume 4 Number 1 (1983) (© AAAI)

Towards a Taxonomy Of Problem Solving Types

B. Chandrasekaran

Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210 USA

Abstract

Our group's work in medical decision making has led us to formulate a framework for expert system design, in particular about how the domain knowledge may be decomposed into substructures We propose that there exist different problem-solving types, i e , uses of knowledge, and corresponding to each is a separate substructure specializing in that type of problem-solving Each substructure is in turn further decomposed into a hierarchy of specialists which differ from each other not in the type of problem-solving, but in the conceptual content of their knowledge; e g , one of them may specialize in "heart disease," while another may do so in "liver," though both of them are doing the same type of problem-solving Thus ultimately all the knowledge in the system is distributed among problem-solvers which know how to use that knowledge This is in contrast to the currently dominant expert system paradigm which proposes a common knowledge base accessed by knowledge-free problem-solvers of various kinds In our framework there is no distinction between knowledge bases and problem-solvers: each knowledge source *is* a problem-solver We have so far had occasion to deal with three generic problem-solving types in expert clinical reasoning: diagnosis (classification), data retrieval and organization, and reasoning about consequences of actions In a novice, these expert structures are often incomplete, and other knowledge structures and learning processes are needed to construct and complete them

This is a revised and extended version of an invited talk entitled, "Decomposition of Domain Knowledge Into Knowledge Sources: The MDX Approach," delivered at the IV National Conference of the Canadian Society for Computational Studies of Intelligence, May 17–19, 1982, Saskatchewan

Introduction

For the past few years our research group has been investigating the issues of problem-solving as well as knowledge organization and representation in medical decision making. In parallel with this investigation we have also been building and extending a cluster of systems for various aspects of medical reasoning. The major system in this cluster is MDX, which is a diagnostic system, i.e., its role is to arrive at a classification of a given case into a node of a diagnostic hierarchy. The theoretical basis of this diagnostic problem-solving is laid out in some detail in Gomez and Chandrasekaran.

The MDX system, which is wholly diagnostic in its knowledge, communicates with two auxiliary systems, PATREC and RADEX. PATREC is a data base assistant in the sense it acquires the patient data, organizes them, and answers the queries of MDX concerning the patient data. In all these activities PATREC uses various types of inferential knowledge embedded in an underlying conceptual model of the domain of medical data. RADEX is a radiology consultant to MDX, and it suggests or confirms diagnostic possibilities by reasoning based on its knowledge of imaging procedures and relevant anatomy. See Mittal and Chandrasekaran (Mittal, Chandrasekaran, 1981) and Chandrasekaran et al (Chandrasekaran, Mittal and Smith, 1980) for further details about these subsystems.

Though in a sense RADEX and PATREC can both be viewed as "intelligent" data base specialists, RADEX has some additional features of interest due to the perceptual nature of some of its knowledge. However, for the purpose of this paper, it is not necessary to go into RADEX in much detail, and we can view PATREC as prototypical of this class of auxiliary systems.

Our aim in this paper is to outline a point of view about how a domain gets naturally decomposed into substructures each of which specializes in one type of problem-solving. Each of these substructures in turn further gets decomposed into small knowledge sources of the same problem-solving type, but specializing in different concepts in the domain. We shall see that this sort of decomposition results in more natural control and focus properties of the overall system. Identification of these substructures and how they communicate with one another is vital to the proper *organization* of the body of knowledge for problem-solving in that domain.

Our method in this paper will be to examine how knowledge is used in a few well-defined tasks: diagnosis, data storage and retrieval, and reasoning about consequences of actions. It should be emphasized that these tasks are not particular to the medical domain. Rather they are fundamental generic tasks occurring in a wide variety of problem-solving situations. Thus these tasks are elements of a taxonomy of basic problem-solving types. When we are done with this examination, the general principles of knowledge decomposition will begin to take on some clarity.

One final point: we will use examples from both medical and non-medical domains. In particular, there are many similarities between reasoning about diseases and therapies on one hand and trouble-shooting and synthesis of corrective actions in complex engineering systems on the other.

The Diagnostic Task

By the term "diagnostic task," we mean something very specific: the identification of a case description with a specific node in a pre-determined diagnostic hierarchy. For the purpose of current discussion let us assume that all the data that can be obtained are already there, i.e., the additional problem of launching exploratory procedures such as ordering new tests etc. does not exist The following brief account is a summary of the more detailed account given in (Gomez, Chandrasekaran, 1981) of diagnostic problem-solving.

Let us imagine that corresponding to each node of the classification hierarchy alluded to earlier we identify a "concept." The total diagnostic knowledge is then distributed through the conceptual nodes of the hierarchy in a specific manner to be discussed shortly. The problem-solving for this task will be performed top down, i.e., the top-most concept will first get control of the case, then control will pass to an appropriate successor concept, and so on. In the medical example, a fragment of such a hierarchy might be as shown in Fig. 1

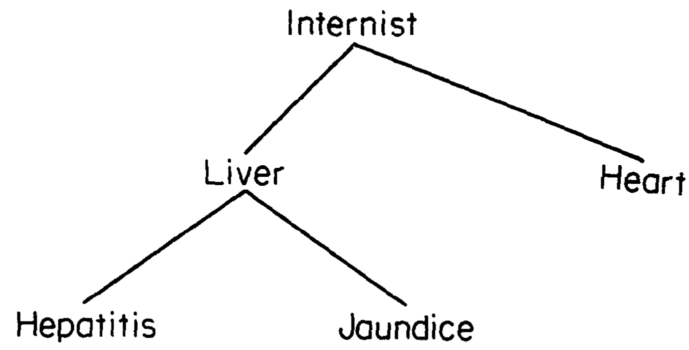

Figure 1

More general classificatory concepts are higher in the structure, while more particular ones are lower in the hierarchy. It is as if INTERNIST first establishes that there is in fact a disease, then LIVER establishes that the case at hand is a liver disease, while say HEART etc. reject the case as being not in their domain. After this level, JAUNDICE may establish itself and so on.

Each of the concepts in the classification hierarchy has "how-to" knowledge in it in the form of a collection of *diagnostic rules* These rules are of the form: <symptoms> → <concept in hierarchy>, e.g., "If high SGOT, add n units of evidence in favor of cholestasis." Because of the fact that when a concept rules itself out from relevance to a case, all its successors also get ruled out, large portions of the diagnostic knowledge structure never get exercised. On the other hand, when a concept is properly invoked, a small, highly relevant set of rules comes into play.

The problem-solving that goes on in such a structure is *distributed.* The problem-solving regime that is implicit in the structure can be characterized as an *"establish-refine"* type. That is, each concept first tries to establish or reject itself. If it succeeds in establishing itself, the refinement process consists of seeing which of *its* successors can establish itself. Each concept has several clusters of rules: confirmatory rules, exclusionary rules, and perhaps some recommendation rules. The evidence for confirmation and exclusion can be suitably weighted and combined to arrive at a conclusion to establish, reject or suspend it. The last mentioned situation may arise if there is not sufficient data to make a decision. Recommendation rules are further optimization devices to reduce the work of the subconcepts. Further discussion of this type of rules is not necessary for our current purpose.

The concepts in the hierarchy are clearly not a static collection of knowledge. They are active in problem-solving. They also have knowledge only about establishing or rejecting the relevance of that conceptual entity. Thus they may be termed "specialists," in particular, "diagnostic specialists " The entire collection of specialists engages in distributed problem-solving.

The above account of diagnostic problem-solving is quite incomplete. We have not indicated how multiple diseases can be handled within the framework above, in particular when a patient has a disease secondary to another disease. Gomez has developed a theory of diagnostic problem-solving which enables the specialists in the diagnostic hierarchy to communicate the results of their analysis to each other by means of a *blackboard* (Erman, Lesser, 1975), and how the problem-solving by different specialists can be coordinated. See (Gomez, Chandrasekaran, 1981) for details. Similarly, how the specialists combine the uncertainties of medical data and diagnostic knowledge to arrive at a relatively robust conclusion about establishing or rejecting a concept is an important issue, for a discussion of which we refer the reader to (Chandrasekaran, Mittal and Smith, 1982).

The points to notice here are the following. The control transfer from specialist to specialist is akin to the corresponding situation in the medical community. We shall have more to say about this later on. Especially note that there is no "problem-solver" standing outside, *using* a knowledge base. The hierarchy of diagnostic specialists is the problem-solver as well as the knowledge-base, albeit of a limited type and scope. That is, the particular kind of problem-solving is *embedded* in each of the units in the knowledge structure.

Data Retrieval and Inference

Consider the following situation that might arise in diagnostic problem-solving that was discussed earlier. Suppose a rule in the liver specialist was: "If history of anesthetic exposure, consider hepatitis " This is a legitimate diagnostic rule in the sense described earlier, i.e., it relates a manifestation to a conceptual specialist. However, suppose there is no mention of anesthetics in the patient record, but his history indicates recent major surgery. We would expect a competent physician to infer possible exposure to anesthetics in this case and proceed to consider hepatitis. Similarly, if a diagnostic rule has "abdominal surgery" as the datum needed to fire it, but the patient record mentions only biliary surgery, it does not take a deep knowledge of medicine to fire that diagnostic rule. In both these cases domain knowledge is needed, but the reasoning involved is not diagnostic reasoning in our specific technical sense. One can imagine an expert diagnostician turning, in the course of her diagnostic reasoning, to a nurse in charge of the patient record and asking if there was evidence of anesthetic exposure or of abdominal surgery, and the nurse answering affirmatively in both the instances without his being trained in diagnosis at all

When we faced this problem in the design of MDX, we realized that it would be very inelegant to combine reasoning of this type with the diagnostic reasoning that we had isolated as a specific type of problem-solving activity. We were led to the creation of a separate subsystem for managing patient data, much like the nurse alluded to earlier. For all questions concerning manifestations, MDX simply turned to this subsystem, which performed the relevant reasoning and returned the answer We were surprised to discover that all the retrieval activities of this "data base assistant" could be captured in a uniform paradigm, to be elaborated

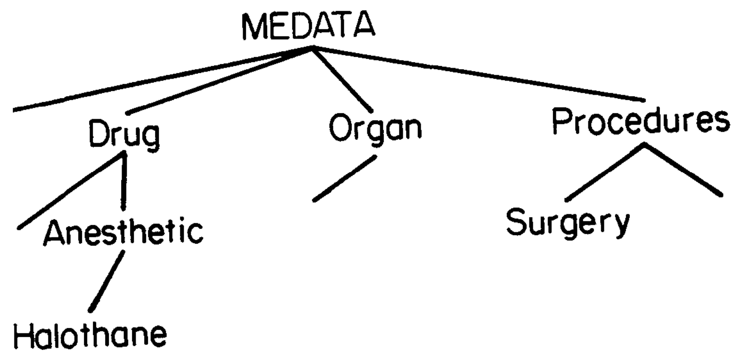

Figure 2

shortly. Mittal (Mittal, 1980) describes this in detail as do the references (Mittal, Chandrasekaran, 1981) and (Mittal, Chandrasekaran, 1969). Similar to our discussion regarding the diagnostic task, we just touch upon the issues here sufficient to make our main points regarding decomposition.

This data base–called PATREC–is organized as a hierarchy of medical data concepts. A fragment of the hierarchy is shown in Fig. 2

At a representational level, there is nothing novel here: each medata concept is represented as a frame, and the inference rules that we will describe shortly are implemented as "demons" or "procedural attachments." However what will be worth noticing is the fact that all these rules will be of a certain uniform type. For the purpose of illustration, let us consider the SURGERY concept. SURGERY frame has LOCATION and PERFORMED? slots, among others. The "PERFORMED?" slot has the following rules:

1. If no surgery in the enclosing organ, surgery not done.
2 If surgery in a component, infer surgery in this organ.
3. If no surgery in any of the components, then infer no surgery in this organ.
4 If evidence of anesthetic, infer "possibly "

The DRUG frame has the following rules in the GIVEN? slot:

1. If any drug of this type given, infer this drug also
2. If the drug class was not given, rule out this particular drug.
3 If all *drugs* of this type were ruled out, rule out the class too

These rules need not be attached to the successors of DRUG, since they can inherit these rules–this is a fairly standard thing to do in frame-based systems. A successor may have further rules which are particular to it, e.g. the ANESTHETIC concept has the rule:

If major surgery, infer ANESTHETIC given, possibly

Let us reemphasize that the interesting thing about the system is not

rare knowledge base system that doesn't—but that it is a collection of conceptual specialists tuned to a particular type of problem-solving. All the embedded inference rules have a common structure: derive the needed data value from data values relating to other concepts. The inferential knowledge that is encoded in the concepts is specific to the data retrieval task in a data base activity.

Let us consider some examples Suppose the stored datum is that "Patient was given halothane." The HALOTHANE frame now has its GIVEN? slot filled with "Yes." Consider the following series of questions:

Q1 Given Anesthetic
A : YES

(ANESTHETIC specialist inherits the rules from the DRUG frame Rule 1 generates the question, among others, "Given Halothane?" "Yes" is propagated upwards)

Q2 Any Surgery performed?
A : Possibly

(SURGERY specialist fails with rules 1, 2 and 3 Rule 4 places query "Given Anesthetic?" to ANESTHETIC specialist "Yes" answer results in "Possibly" to Q2 This is an example of lateral inheritance)

Similarly if the stored datum were "Patient had major surgery," and the query were, "Given Anesthetic?", rule 1 in ANESTHETIC would have given the answer "possibly."

Another more complex example of data retrieval reasoning by PATREC is the following:

DATA: A liver-scan showed a filling defect in the left hepatic lobe. The liver was normal on physical exam
Q : Liver Normal?
A : No

(On liver-scan data, the following chain of inference took place: (a) filling-defect in lobe → lobe not normal; (b) If <comp-of> liver not normal → liver not normal. On the other hand, Physical examination produced "Normal" as answer. By using a general principle that when there are contending answers, *non-default* value should be chosen—the default for "Normal?" slot of LIVER is "Yes"—the answer "No" was generated)

The main points relevant here are, as in the case of the diagnostic task: (1) There is no separation between a knowledge base and a problem-solver Problem-solving is embedded in the knowledge structure. (2) All the conceptual specialists perform the *same type* of problem-solving, in this case, inheritance of data from other specialists. (3) Concepts with the same name, say LIVER, in the diagnostic structure and the data retrieval structure have different pieces of knowledge and do different things This is akin to the fact that the LIVER concept of a diagnostician is bound to be different from that of the data base nurse. The concepts in this sense are "tuned" for different types of knowledge use.

What-Will-Happen-If (WWHI) Or Consequence Finding

We said that among the many types of problem-solving that take place in a knowledge-rich domain is that of answering questions of the form "What will happen if X is done?" Examples are: "What will happen if valve A is closed in this power plant when the boiler is under high pressure?"; "What will happen if drug A is administered when both hepatitis and arthritis are known to be present?" Questions such as this can be surprisingly complex to answer since formally it involves tracing a path in a potentially large state space. Of course what makes possible in practice to trace this path is domain knowledge which constrains the possibilities in an efficient way.

The problem-solving involved, and correspondingly the use of knowledge in this process, are different from that of diagnosis For one thing, many of the pieces of knowledge for the two tasks are completely different. For example, consider answering the question in the automobile mechanic's domain: "What will happen if the engine gets hot?" Looking at all the diagnostic rules of the form, "hot engine → <malfunction>" will not be adequate, since <malfunction> in the above rules is the *cause* of the hot engine, while the consequence finding process looks for the effects of the hot engine Formally, if we regard the underlying knowledge as a network connected by cause-effect links, where from each node multiple cause links as well as effect links emanate, we see that the search processes are different in the two instances of diagnosis and consequence-finding The diagnostic concepts that typically help to provide *focus* and constrain search in the pursuit of correct causes will thus be different from the WWHI concepts needed for the pursuit of correct effects.

The embedded problem-solving is also correspondingly different. We propose that the appropriate language in which to express the consequence-finding rules is in terms of *state-changes* To elaborate:

1 WWHI -condition is first understood as a state change in a subsystem.
2. Rules are available which have the form "<state change in subsystem> will result in <state change in subsystem>" Just as in the case of the diagnosis problem, there are thousands of rules in the case of any nontrivial domain Again, following the diagnostic paradigm we have already set, we propose that these rules be associated with *conceptual specialists* Thus typically all the state change rules whose left hand side deals with a subsystem will be aggregated in the specialist for that subsystem, and the right hand side of those rules will refer to the state changes of the *immediately affected* systems

Again we propose that typically the specialists be organized hierarchically, so that a subsystem specialist, given a state change to it, determines by knowledge-based reasoning the state changes of the immediately larger system of which it is a part and calls that specialist with the information determined by it. This process will be repeated until the state change(s) for the overall system, i e , at the most general relevant level of abstraction, are determined. This

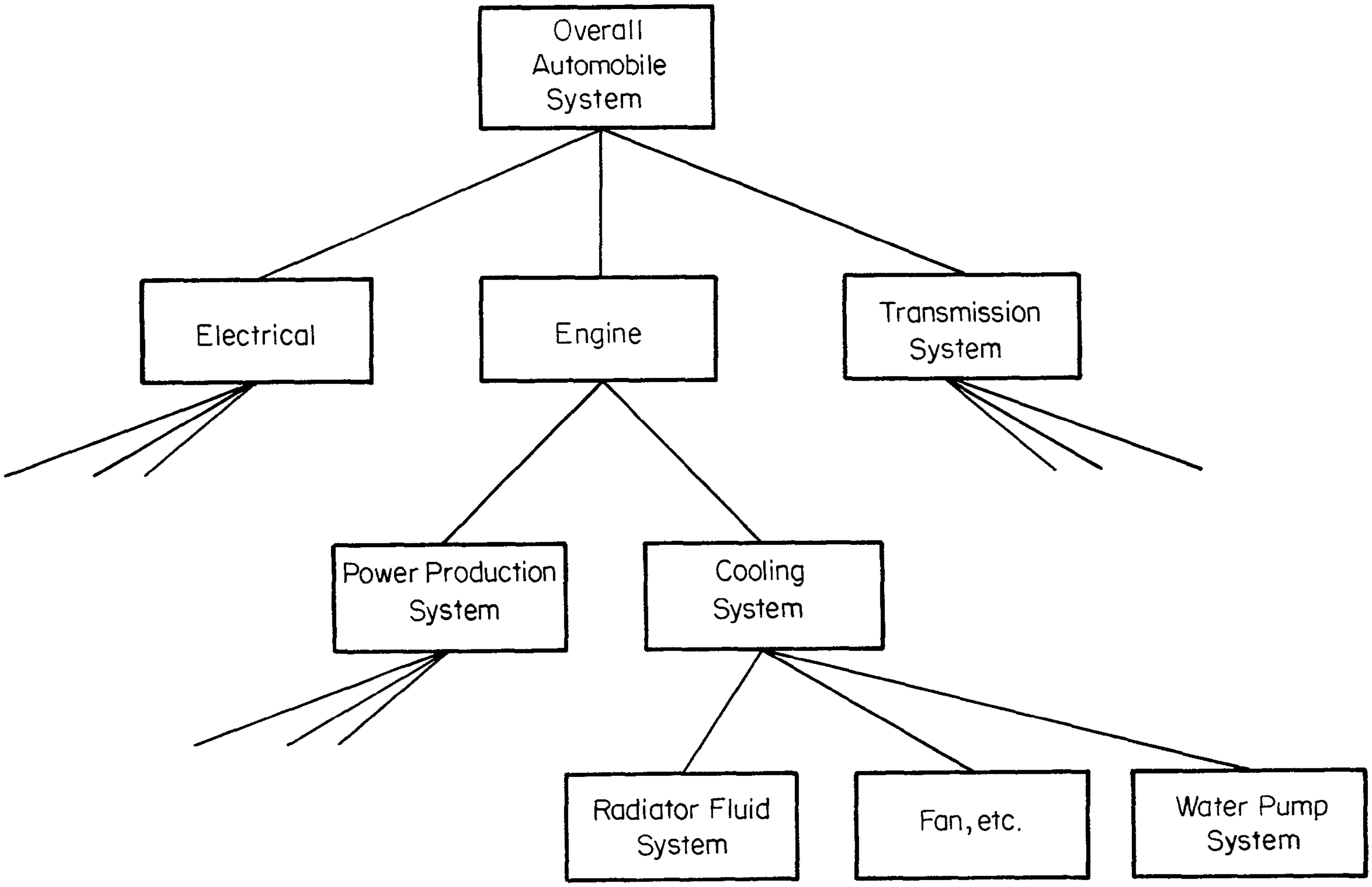

Figure 3

form of organization of the rules should provide a great deal of focus to the reasoning process.

An Illustrative Example. Consider the question, in the domain of automobile mechanics, "WWHI there is a leak in the radiator when the engine is running?" We suggest the specialists are to be organized as in Fig 3.

The internal states that the *radiator fluid subsystem* might recognize may be partially listed as follows: {leaks/no leaks, rust build-up, total amount of water,...}; similarly, the *fan subsystem* specialist might recognize states {bent/straight fan blades, loose/tight/disconnected fan belt,...}. The *cooling system subsystem* itself need not recognize states to this degree of detail; being a specialist at a somewhat higher level of abstraction it will recognize states such as {fluid flow rate, cooling-air flow rate...etc.}. Let us say that the *radiator fluid* specialist has, among others, the following rules. The rules are typically of the form:

<internal state change> → <supersystem state change>

leak in the radiator → reduced fluid flow-rate
high rust in the pipes → reduced fluid flow-rate
no antifreeze in the water and very cold weather → zero fluid flow etc.

The cooling system specialist might have rules of the form:

low fluid-flow rate and engine running → engine state hot
low air-flow rate and engine running → engine state hot

Again note that the internal state recognition is at the appropriate level of abstraction, and the conclusions refer to state changes of its parent system.

It should be fairly clear how such a system might be able to respond to the query about radiator leak. Again a blackboard for this task would make it possible to take into account subsystem interaction.

Unlike the structures for the diagnostic and data retrieval tasks, we have not yet implemented a system performing the WWHI-task. While we cannot speak with assurance about the adequacy of the proposed solution, we feel that it is of a piece with the other systems in pointing to the same set of morals: embedding still another type of problem-solving in a knowledge structure, which consists of cooperating specialists of the same problem-solving type

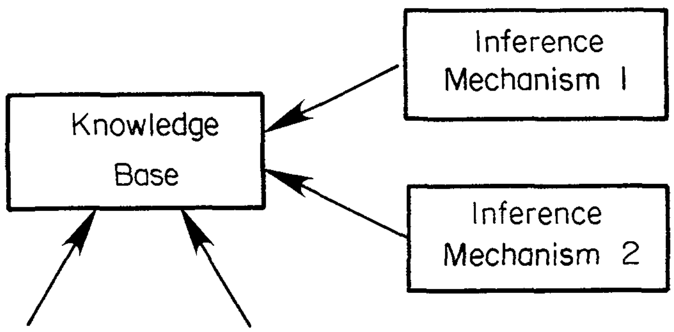

Figure 4

Knowledge-Use Taxonomy

There has been a growing realization in the field that the important issue in knowledge systems is to determine how knowledge is to be used. Our foregoing examination of the three tasks—each of which is not some ad hoc need for medical reasoning, but is a generic task that arises in a number of domains—leads us to propose the following theses.

1. There is taxonomy of problem-solving regimes that are involved in expert problem-solving. We have identified three members of this taxonomy
   • diagnostic (classificatory): establish-refine, top-down.
   • consequence-finding: abstract state from low-level description to higher-level description, bottom-up.
   • data retrieval: inheritance/inference of values from data values in other concepts.
   There are obviously more. Our research is oriented towards finding more elements of this taxonomy and determining their interrelationships
2. For each type of problem-solving there is a separate knowledge structure, with the associated p.s. regime embedded in it. Thus a domain of knowledge can be decomposed into a collection of structures, each of which specializes in a p.s. type. We can call this a horizontal decomposition of the domain.
3. Each of the structures in (2) above can be further decomposed into a collection of specialists, all of whom are of the same p.s. type, but differ from each other in the conceptual content We have indicated how this decomposition can be done for the three tasks considered. We term this decomposition a vertical decomposition.

A Paradigm Shift

The prevalent approach to knowledge base systems is based on the decomposition in Fig 4:
In this paradigm, knowledge representation is separated from its use. This approach has the attraction of generality and a certain kind of modularity.

The representational questions are dealt with in this approach in a manner to satisfy the criterion of expressiveness, or so-called epistemological adequacy of McCarthy (McCarthy, Hayes, 1969). The efficiency responsibilities are put on the shoulders of the inference mechanisms; they have to have the so-called heuristic knowledge in order to use the knowledge efficiently for problem-solving. Our approach is based on a rather different decomposition of the same problem, as indicated in our discussion on horizontal decomposition in the previous section.

Pictorially, the viewpoint of knowledge-based systems that we advance can be given as Fig. 5.

Thus the overall knowledge system is viewed as a *collection of specialists in inference types,* who cooperatively solve a given problem. While in the figure we have indicated the communication among these specialists to be unconstrained, in fact, however, it may not be so. There may be reasons why only certain problem-solving specialists can talk to other problem-solving specialists. This is an open research problem in our approach.

Production Rule Methodology. In most of the preceding discussions the *representation* of knowledge has been in the form of rules. We feel that this is not accidental, but that rules represent a basic form of cognition, viz., "how-to" knowledge. This was recognized early in AI by Newell and Simon (Newell, Simon, 1972) who named the rules *production rules.* Later, the Stanford Heuristic Programming Project and others extended this production rule methodology for a wide class of expert system design problems. We are thus in agreement with the use of rules as a basic knowledge representation formalism in expert systems.

There are two aspects in which our methodology differs from current work on rule based systems. We have already alluded to the difference in the viewpoint which regards knowledge not as an independent structure to be used by different problem-solvers, but as *embodiments* of implicit problem solving knowledge. Related to that is the idea that the central determinant of effective use of knowledge is how it is *organized* Our approach begins to provide criteria for performing the organization of a complex body of knowledge. It is well-known that production rules need to be organized not simply for purpose of efficiency, but for *focus* and *control* in problem-solving (see (Lenat, Harris, 1978) for a discussion of these issues). We are proposing two *organizing constructs,* which extend the production rule methodology to make it applicable to a larger class of problems. One construct is the problem-solving regime and the other is that of a *conceptual specialist*

Related to these organizational notions is the other aspect of the difference between our approach and the current production rule methodologies. We do not use uniform problem-solving mechanisms (backward chaining, e.g.) across the whole domain. As indicated, the problem-solving method differs from knowledge structure to knowledge structure.

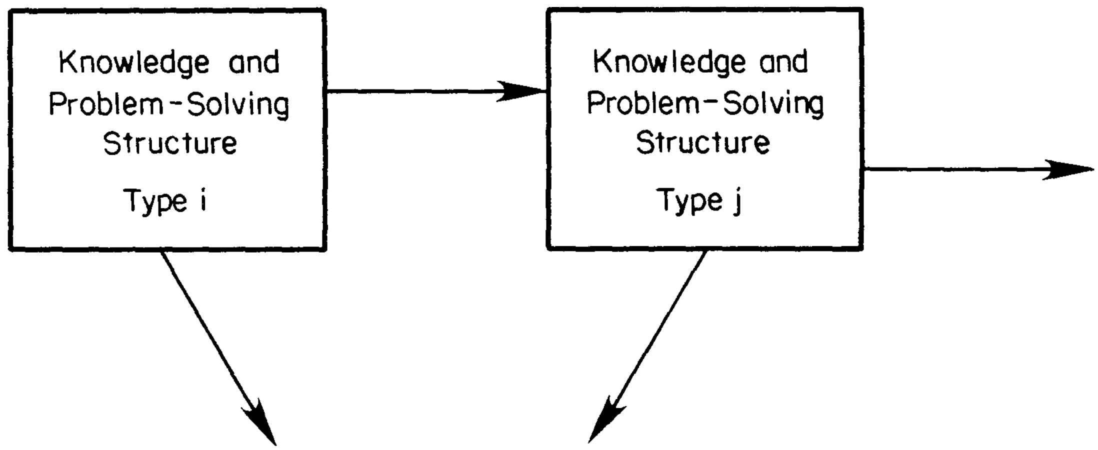

Figure 5

Role of "Deep" Models

Deep and Compiled Structures. Recently Hart (Hart, 1982) and Michie (Michie, 1982) have written about the "depth" at which knowledge is represented and used in problem solving by expert systems Distinctions such as "deep" vs "surface" and "high road" vs "low road" have been made in this connection. There is no clear definition of what constitutes a deep model – in fact precisely that issue is an open area of research in the field, but the intuition is that it models the underlying processes of the domain. Michie remarks that most expert systems that are extant don't have deep models in this sense, but instead can be viewed as a data base of patterns with a more or less simple control structure to navigate through the data base. It is argued that surface systems of this type have inherent limitations in hard problems, and that a system which has a deep model will be able to turn to it when faced with an especially knotty problem, much like a human expert might resort to "first principles" in a similar situation. In addition to deep models of the domain, the human problem solver also uses other sorts of knowledge such as "common sense" knowledge of various kinds.

In the rest of the discussion in this section we will explicitly consider the diagnostic task only. But the arguments will apply to other tasks as well.

We argue in (Chandrasekaran, Mittal, 1982) for a thesis which might at first sound counter-intuitive. Let us assume that we wish to design a diagnostic system in a particular domain. Let us further assume that we can successfully construct a deep model of the domain, and also specify the problem solving processes that will operate on that model. The thesis that we argue in (Chandrasekaran, Mittal, 1982) is as follows. Between the extremes of a data base of patterns on one hand and representations of fdeep knowledge (in whatever form) on the other, there exists a knowledge and problem solving structure – along the lines outlined in the section on the diagnostic task in this paper – which (1) has all the relevant deep knowledge "compiled" into it in such a way that it can handle all the diagnostic problems that the deep knowledge, if explicitly represented and used in problem-solving, can handle; and (2) will solve the diagnostic problems more efficiently than the deep structure can; but (3) it cannot solve other types of problems– i.e , problems which are not diagnostic in nature – that the deep knowledge structure potentially could handle. The argument is rather detailed, but the essence of it consists of analyzing the ways in which the diagnostic structure may fail to solve a particular problem, and tracing that failure to either missing knowledge in the deep model itself or in the problem solving processes that operate on it. Thus the range of *diagnostic* problems that can be solved with the deep model is exactly coextensive with the problems solvable with the diagnostic problem solving structure that can be derived from it.

There is another way of looking at this. There is a natural decomposition in the problem solving responsibilities between the underlying knowledge structures and the diagnostic structure. The former builds the diagnostic structure and the latter solves specific diagnostic problems. Human experts often resort to deep models because the diagnostic structures are in general incomplete This decomposition also translates into a natural division of responsibility for explanation of decisions. See (Chandrasekaran, Mittal, 1982) for more discussion on this.

Multiple Uses of Knowledge. It is possible that there will be some redundancy in knowledge represented in our approach, since it calls for knowledge to be encoded in a problem solving structure according to its usage – some pieces of knowledge may appear in several structures (See comments in (Gomez, Chandrasekaran, 1981) on redundancy and biasing of knowledge.) Is this a good thing?

We have a choice: (1) We can have the knowledge in a deep enough form, but as, say, a diagnostic problem presents itself, we can first generate fragments of diagnostic knowledge as needed and use it to solve the given problem. Similarly for a WWHI problem, etc. Or, (2) we can choose the tasks to be experts in , compile the problem solving structures for them, accepting some redundancy. The latter is faster for those tasks for which they are designed, the former is more economical in storage. A classic trade-off!

In a sense the former situation describes, e.g., a bright medical school graduate who has a functional understanding of the phenomena of the human body, but that knowledge is not yet molded into effective problem solving structures of particular types. We suspect that what happens even among experts is that they build powerful problem solving structures to account for a good portion of foreseeable situations, and thus need to resort to the deeper structures only for the harder problems. This is a compromise between the requirements of expertise and memory.

The Nature of the Deep Model

There is an additional problem with option 1 in the current state of the art: we don't know how to do it! This requires an adequate theory of the nature of the deep model. When a person newly understands how a device works, e.g., it is doubtful that what he has acquired is merely a collection of rules or facts, or a network of causal relations. One can have all these and still not "understand." The sense of understanding must correspond to some *organization* of these pieces of knowledge for some class of purposes. The organization must be such that it can be processed to produce problem solving structures for various tasks. The nature of the deep model is an extremely important area of research. The work of (Rieger, Grinberg, 1976), (Pople, 1982), (Patil, 1981) and (de Kleer, Brown, 1982), to name a few researchers who have looked at this problem, seem very relevant here. However, in order to adequately represent knowledge at this level, notions of an organizational nature particular to that level also seem important.

On Hierarchies

In all the tasks that we considered in this paper, the knowledge structures were strongly hierarchical. While hierarchical organizations have a strong intuitive appeal, in AI there is also a strong tradition of "heterarchies" and network structures. Difficulties with hierarchical classification structures have been noted in (Fahlman, et al, 1981). Also concerns such as "the world is not hierarchical" are voiced in response to proposals for hierarchical organizations.

This is not the place to discuss the important issue of hierarchical structures in problem solving. The following brief remarks should suffice for our purposes. First of all, the main thesis about decomposing knowledge by problem solving types and embedding of the problem solving in the knowledge sources is itself independent of whether the structures for a problem solving type are hierarchical. Secondly, our general strategy has been to start by looking for hierarchical decompositions, and where there seems to be a need for communication outside of the hierarchical channels, to provide it in a carefully controlled fashion such as the blackboards discussed in (Gomez, Chandrasekaran, 1981). (See (Chandrasekaran, 1981) for a discussion of different kinds of communication needs in a distributed problem solving situation.) For example, in (Gomez, Chandrasekaran, 1981) we discuss how certain kinds of relations between disease hypotheses belonging to different portions of the hierarchy – such as disease A being secondary to disease B – can be handled within a hierarchical framework by the use of blackboards. Finally, it ought to be stated clearly that hierarchies are not "out there," but imposed by the thought processes for control over problem solving. Thus it is a powerful weapon, but by no means a sufficient one. It will be rash to conclude that all complex problem solving in all complex domains can be crisply conducted in a single hierarchical framework. Reasoning about feedback and reasoning with multiple perspectives are two examples where additional machinery seems to be needed beyond the hierarchical framework.

The Organization of the Medical Community

Evidence of Horizontal Decomposition. The medical community collectively is a good case study in the principles by which knowledge may be structured for cooperative, effective problem-solving Corresponding to our notion of horizontal decomposition along the lines of problem-solving types, we can identify clinicians, educators, pathologists, radiologists, medical records specialists, etc. Clinicians combine the diagnostic and predictive knowledge structures, for practical reasons having to do with the close interaction between diagnosis and therapy. Medical record specialists, as their name indicates, serve to organize patient data and retrieve them effectively. Radiologists are not diagnosticians in the same sense as clinicians are: their problem-solving is to reason from imaging descriptions to confirm or reject diagnostic possibilities; they are largely perceptual specialists.

Evidence of Vertical Decomposition. Corresponding to our vertical decomposition, many of the above problem-solvers are organized into conceptual hierarchies. For instance, an internist is the top-level diagnostic specialist, who may call upon liver or heart specialists for further investigation of a problem. The top-down problem-solving for diagnosis is indicated by the fact that a sick person typically first goes to an internist, who may refer the patient on to more detailed specialists.

Evidence for Embedding Problem-Solving. If the medical community were organized according to the currently accepted paradigm in expert systems, i.e , a common knowledge base shared by different problem-solvers who themselves are without domain-knowledge, one would expect to have knowledge-specialists, who would be rather like encylopaedias, and problem-solving specialists who would possess expert-algorithms for, say, diagnosis, without any medical knowledge about particular medical concepts. Thus whenever a patient came, the diagnostic specialist would consult the knowledge-base specialist and together they would arrive at a diagnostic conclusion.

However, that is not the way the community works. Instead we find that experienced medical specialists possess expertise which is not a raw knowledge-base, but which is highly effective in problem-solving. On the other hand, a medical student without clinical experience is more like a pure knowledge-base As he or she becomes more experienced in various types of problem-solving, the unstructured knowledge base slowly begins to shape and structure itself, so that pieces of knowledge are tuned for ready and effective use.

Acknowledgments

I owe a great debt of intellectual gratitude to Fernando Gomez, who was responsible for many of the central theoretical ideas behind MDX Sanjay Mittal has been a dedicated co-worker, responsible for much of the implementation as well as for many of the ideas incorporated in PATREC and RADEX. Jack Smith, M.D., is, in addition to being the medical component of our group, also a skilled computer scientist, which contributed greatly to a smooth AI/medical interface. I thank Lee Erman who read a draft of this paper and made several suggestions to improve its clarity. Needless to say his help does not necessarily imply endorsement of all the ideas in this paper.

The preparation of this paper was supported by NSF grant MCS-8103480 and AFOSR grant 82-0255. The computing resources of Rutgers University Laboratory for Computer Science Research were used extensively in system development, and this was made possible by the Biotechnology Resources Division of NIH, under grant RR-00643.

References

Chandrasekaran, B. (1981) Natural and social system metaphors for distributed problem solving: Introduction to the issue *IEEE Transactions on Systems, Man & Cybernetics* SMC-11(1):1-5.

Chandrasekaran, B. and Mittal, S. (1982) Deep versus compiled knowledge approaches to diagnostic problem-solving *Proc of AAAI,* 349-354.

Chandrasekaran, B , Mittal, S., and Smith, J (1980) RADEX—Towards a computer-based radiology consultant. In E S. Gelsema and L. N. Kanal (Eds.), *Pattern Recognition in Practice* Amsterdam, North Holland.

Chandrasekaran, B., Mittal, S., and Smith, J. W. (1982) Reasoning with uncertain knowledge: the MDX approach. *Proceedings of the 1st Annual Joint Conferences of the American Medical Informatics Association.*

Erman, L. D. and Lesser, V. R. (1975) A multi-level organization for problem-solving using many diverse cooperating sources of knowledge. *IJCAI 4,* 483-490.

Fahlman, S. E , Touretzky, D. S., and van Roggen, W (1981) Cancellation in a parallel semantic network. *IJCAI 7*

Gomez F. and Chandrasekaran, B. (1981) Knowledge organization and distribution for medical diagnosis. *IEEE Transactions on SMC,* SMC-11:34-42.

Hart, P. E. (1982) Direction for AI in the eighties. *SIGART Newsletter,* 79:11-16.

de Kleer, J. and Brown, J S. (1982) Foundations of envisioning. *Proc. of AAAI,* 434-437.

Lenat, D B. and Harris, G (1978) Designing a rule system that searches for scientific discoveries. In D A. Waterman and F. Hayes-Roth, (Eds.), *Pattern-Directed Inference Systems* New York: Academic Press.

McCarthy, J. and Hayes, P. J. (1969) Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence,* Vol. 4, New York: Elsevier.

Michie, D. (1982) High-road and Low-road programs *AI Magazine,* 3:21-22

Mittal, S. (1980) *Design of a distributed medical diagnosis and database system* Doctoral Dissertation, Dept. of Computer and Information Science, The Ohio State University.

Mittal S. and Chandrasekaran, B. (1981) Conceptual representation of patient data bases. J Medical Systems

Mittal, S. and Chandrasekaran, B. (1981) Software design of knowledge-directed database systems. *Proc. 1st Conf Foundations of Software Technology and Theoretical Computer Science,* Tata Institute of Fundamental Research, Bombay, India

Mittal, S. and Chandrasekaran, B. (1981) Some issues in the design of knowledge-directed databases Working Paper, AI Group, Dept of Computer and Information Science, The Ohio State University.

Newell, A. and Simon, H A. (1972) *Human Problem Solving* Englewood Cliffs, NJ: Prentice-Hall

Patil, R S. (1981) *Causal representation of patient illness for electrolyte and acid-base diagnosis* Doctoral Dissertation, TR-267, MIT Lab for Computer Science, Cambridge, Mass

Pople, H E (Forthcoming) Heuristic methods for imposing structure on ill-structured problems. In P. Szolovits (Ed), *Artificial Intelligence in Medicine* Westview Press

Rieger, C. and Grinberg, M (1976) The causal representation and simulation of physical mechanisms Tech Rep TR-495, Dept of Computer Science, The University of Maryland.