CONTROL AND INTEGRATION OF DIVERSE KNOWLEDGE IN A DIAGNOSTIC EXPERT SYSTEM

Pamela K. Fink

Southwest Research Institute
6220 Culebra Rd.
San Antonio, Texas 7 8284

ABSTRACT

Though current expert system technology has become a major success, existing expert systems often fall short of human expertise In many ways. One Important area is in the use of more basic, deep knowledge as an enhancement to the shallow, surface knowledge commonly employed. The Integrated Diagnostic Model attempts to exploit the use of both types of knowledge by fitting the appropriate knowledge representation and utilization techniques to each. The result is two separate and Independent expert systems which are then integrated and controlled by a higher-level module called the executor.

I INTRODUCTION

In solving a problem, human experts often draw on many different kinds of knowledge. What usually makes an expert an expert Is the "shallow" [6], "surface" [11], "compiled" [1], "low-road" [9],"emplrical"[2], or "experiential" [4] knowledge. Such knowledge Is acquired from other experts in the field or by first-hand experience in solving domain problems. It consists of rules-of-thumb that connect characteristics of a problem with its possible solutions. However, this type of knowledge is often just a short-cut through, or a compilation of, a more detailed and deeper understanding of the problem domain. This other kind of knowledge, referred to as "deep", "high-road", "functional", or "physical" knowledge, allows an expert to "reason from first principles" in trying to solve a particularly difficult or unfamiliar problem.

The human expert has no apparent difficulty in converting from one kind of knowledge to another. He/she may begin solving a problem by using the experiential knowledge which leads to an isolation of the problem and then convert to using the deeper knowledge to analyze the problem further. If the initial hunch proves to be wrong upon closer examination, the expert easily goes back to employing the more shallow, experiential knowledge. Should all experience fall, he/she can still revert back to the deeper knowledge in attempting to solve the problem.

Though current expert system technology relies heavily on shallow knowledge in solving a problem, some work has been done in developing systems that have deeper knowledge [1,2,3,4,5,7,8,12,13,15]. There are recognized benefits in being able to provide a deeper understanding of the problem, such as better performance at the periphery of the knowledge base and Improved explanation capabilities. However, shallow knowledge must also be available for solving the more common problems and for situations where human understanding of the domain is not extensive enough to include a deeper model. Therefore, It would be best for an expert system to Integrate at least these two often diverse types of knowledge and be able to use them appropriately and efficiently.

The Integrated Diagnostic Model (IDM) [4], shown in Figure 1, is an expert system that contains both a shallow and a deep knowledge base. It was designed to work in the area of mechanical/electronic diagnosis and repair. It consists of three main modules: 1) the experiential expert, 2) the physical expert, and 3) the executor. Knowledge representation and utilization techniques have been developed to fit the two diverse types of knowledge In these specialized mechanical/electronic diagnostic experts. In addition, the higher level control module, the executor, has been designed to direct the entire problem solving process.

II KNOWLEDGE REPRESENTATION IN THE IDM

Each of the three parts of the IDM has its own knowledge base. The knowledge representation techniques used in the experiential and physical experts attempt to depict the type of knowledge that each embodies. The knowledge base In the executor provides a link between the two.

The experiential expert contains a knowledge base implemented as a three-level semantic network where each level is a semantic network in Its own right. Its general structure is shown in Figure 2 and resembles that of Casnet [15]. This structure attempts to model some of the initial experiential, causal reasoning that an expert does while solving a familiar problem. The lowest level in the semantic network, the 'information' level, contains any concrete, observable facts that can be obtained from the user. This could be information on how tight the generator belt is, a voltage measurement in an electronic circuit, or the stain of an organism in a lab culture* The middle level, the hy-

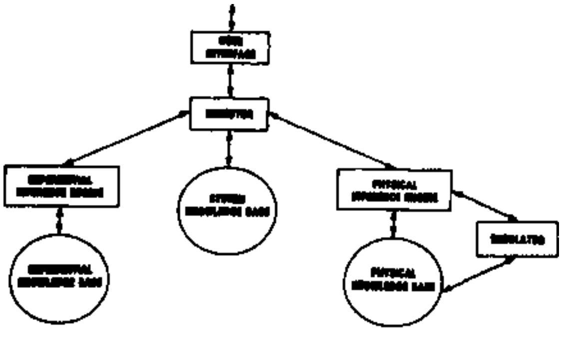

Figure 1. Overview of the Integrated Diagnostic Model

potheses' level, contains knowledge that is not observably apparent but that can be deduced from the observable facts based upon experience. It usually concerns physical or physiological states of the device. Examples of this type of knowledge are that the generator is not recharging the battery, that a wire is shorted in the circuit board, or that the patient is suffering from an infection of pseudomonas bacteria. The 'solutions" level in the semantic network contains knowledge about how to fix the deduced problem. Thus, it would suggest tightening the generator belt, repairing the wire, or placing the patient on a drug therapy of colistin.

The physical expert contains a physical/functional model of the device under diagnosis. This model is based on a set of functional primitives that allows the device under consideration to be simulated qualitatively [8]. Examples of such primitives are:

1) transformer
an object which converts given substances into others; examples are: a generator converting mechanical energy into electricity (mechanics), a solar cell converting photons into electricity (electronics), the collection of enzymes that converts glycogen into carbon dioxide and water (physiology).

2) regulator
an object which controls the activity of another object, usually as a function of the regulator's inputs; examples are: a turbocharger increasing the amount of fuel in the cylinder based upon the pressure of exhaust gasses (mechanics), a transistor controlling the current flow as a function of the base current (electronics), white blood cells regulating the amount of foreign objects in the blood stream (physiology).

3) reservoir
an object which stores a substance for release at a later time; examples are: a spring storing mechanical energy (mechanics), a capacitor storing a charge (electronics), mitochondria storing chemical energy in a cell (physiology).

4) conduit
an object which transports various substances throughout the system; examples are: a piston transferring thrust from a cylinder to the camshaft (mechanics), a wire transporting electricity (electronics), arteries carrying oxygen to cells (physiology).

5) joint
an object which provides a connection between two or more of the other primitives; examples are: the meshing of gears (mechanics), the wire nut that connects two wires (electronics), the synapse between neurons in the brain (physiology).

These primitives are implemented using a semantic network. The first three are represented by frame-like [10] nodes while the last two are represented by arcs.

This set of primitives is by no means complete. For example, it was necessary in the electrical domain to divide 'regulator" into 'switch' and 'relay' because their behavior differences were important in representing how a circuit functions. Such a set provides an integrated way of representing the structure and behavior of a device. Both physical and functional aspects can be represented within this one model. Depending on the substance, such as electricity or water, that is being transported from node to node via the conduits and joints, an electrical, mechanical, etc. representation can be provided.

The system knowledge base for the executor consists of a conversion table that indicates what node or nodes in the experiential knowledge base are associated with what functional primitive nodes or arcs in the physical knowledge base and how they are related. The conversion table is essentially a set of registers that represent the nodes in the experiential knowledge base and the functional primitive nodes and arcs in the physical knowledge

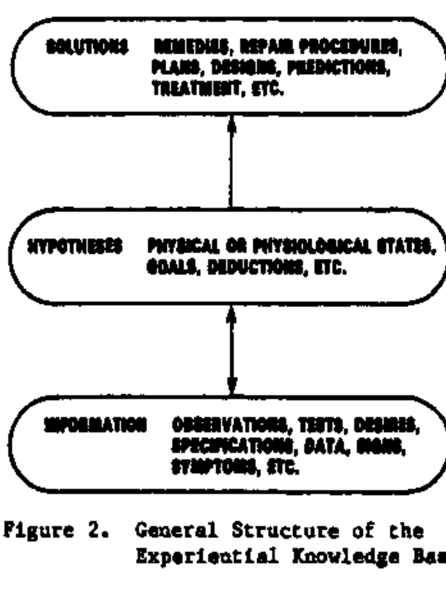

Figure 2. General Structure of the Experiential Knowledge Base

base. These registers contain values that indicate the state of the system's knowledge about that part of the problem. Each register is also associated with a conversion equation or equations that tell tow to convert the information in a register or re* Sisters associated with one knowledge base into information in a register or registers associated with the other knowledge base. The correspondence between registers associated with the two knowledge bases is not necessarily one-to-one, but could be aany-to-one, one-to-many, or many-to-many.

III CONTROL IN THE IDM

Each of the three modules of the IDM has its >wn control mechanism. The inference engines of :he two experts were designed to best take advan-:age of the knowledge available to each. Since the :wo types of knowledge have independent knowledge representations and inference engines, they create completely autonomous expert systems. These two expert systems are then Integrated by the control lechanism of the executor.

The inference engine for the experiential expert is model-driven using a best-first search strategy. Initial data is acquired at the information level which then suggests hypotheses to consider. The hypotheses may then suggest more data that needs to be acquired from the information lev»1. On each arc there is a set of ratings, including probability, cost, and difficulty, that are :ombined to form a confidence factor for the association between the two nodes that the arc conlects. The highest factor is followed first. The search back and forth between hypotheses and information continues until a hypothesis has been verified with enough confidence. Then the arcs between :he hypotheses and solutions are followed. Should :he solution fall to solve the problem, others can be tried or the search for another cause of the >roblem can continue.

The inference engine for the physical expert ses some very general diagnostic reasoning rules :o examine the physical model and Isolate the probLem to a specific functional unit or units, represented by the appropriate functional primitive. These rules are based upon the values of the Inputs and outputs of the functional units. They are:

[) If an output of a functional unit is unknown, ask the user.

2) If an output from a functional unit appears incorrect, check its input.

3) If an input to a functional unit appears incorrect, check the source of the input.

i) If the input of a functional unit appears to be correct, but the output is not, assume that something is wrong with the functional unit being examined.

This resembles the "discrepancy detection" methodology discussed by Davis [2]. Once the problem has been Isolated, more specific analysis can proceed in the same manner at a lower level, since the knowledge representation can be hierarchical.

The control at the executor level determines when each expert system is to be used. It also controls the transfer of information about the state of parts of the device under diagnosis between the expert systems and directs the interaction between the system and the user.

Currently, top level control over which expert to use when is very simplistic. The experiential expert is run first and if it fails to find a solution to the problem, the physical expert is run. This appears to be how an expert would reason about a problem, checking out all reasonable, familiar possibilities first, before resorting to a more basic "first principles" analysis.

A human, however, can use the knowledge gained during the experiential phase to help In the more detailed analysis. At the same time, knowledge acquired while doing a detailed analysis is not lost to the human expert when he/she goes back to the experiential type of reasoning. An expert system should be able to do the same. Thus, while the experiential expert is running, all information gained is passed to the executor for conversion to a form useful to the physical expert, when appropriate. In this way the physical expert can monitor the progress and make certain inferences about the state of the device that is not available to the experiential expert. Through this mechanism, it acquires all of the information that the experiential expert acquires, thus allowing it to begin diagnosis where the experiential expert leaves off, should the need arise. By this same mechanism it can communicate inferences it makes back to the experiential expert. These capabilities will be demonstrated by an example in the next section.

The executor controls communication of information between the user and the two expert systems. Any request for information needed by one of the expert systems is given to the executor which then determines whether the information is already known by the other expert system, or whether to ask the user. If the Information is already available from the other expert system, the executor simply returns the information in the proper form to the requesting expert system. Otherwise it asks the user. Should the user request an explanation at any point in the dialogue, the executor determines which expert should answer it. Currently, only "how" and "why" are legal questions. The answer to "how" is associated with the node that resulted in the last output to the user. The answer to "why" is currently always answered by the physical expert since the experiential expert could only offer a very shallow explanation.

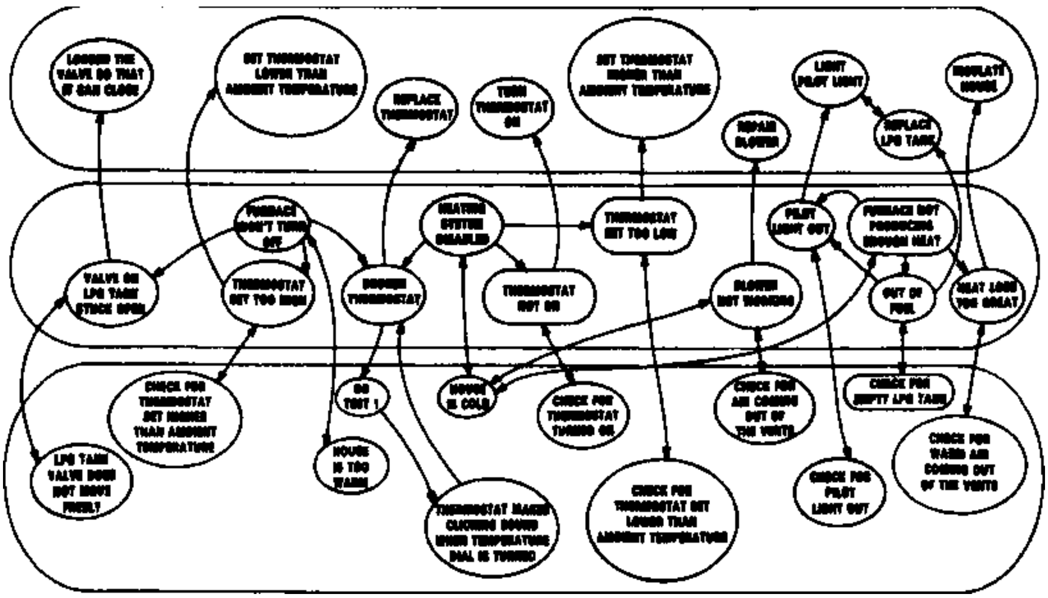

Figure 3. An Example Experiential Knowledge Base for a Simple Heating System

IV AN EXAMPLE

To illustrate how the system works, a simple heating system will be used as an example. The experiential and physical knowledge bases are shown in Figures 3 and 4, respectively. During a session, the physical representation appears in the top two-thirds of the screen to graphically portray the progression of the problem solution. The bottom third contains the dialogue.

Suppose that the user enters the initial problem that the house is too cold. The experiential expert begins by checking out the possibility that the heating system is disabled in some way. This is chosen first because it is very inexpensive and easy to check and it is necessary to know if the heating system is even set to be on. This leads to questions to the user concerning the state of the thermostat and, when it is established that the thermostat is on, set correctly, and working, a register in the executor associated with the experiential knowledge base is set to indicate that the thermostat is alright. This information is useful to the physical expert in a different form. Thus, a conversion takes place that sets registers associated with the power supply, the coil, and the contacts to correct. This causes the nodes on the display screen representing these units to turn dark, giving some indication to the user of the progress being made. This part of the example demonstrates how the system handles a many to many conversion of information. There is no such unit as a thermostat with all of its settings in the physical knowledge base, just as there is no power supply, coil, or contacts in the experiential knowledge base.

The experiential analysis continues by check* ing the blower and then the furnace itself. The first thing that the experiential expert wants to know under the hypothesis that the furnace is not producing enough heat is if the pilot light is on. If the answer to this question is yes, a register associated with the pilot light of the experiential knowledge base is set, which triggers a conversion to set a register associated with the pilot light of the physical knowledge base. In this case, the conversion is one-to-one and therefore straightforward. However, the physical expert can assume from this information that the liquid propane gas (1pg) tank has gas in it (at least as a first cut), that the lpg tank valve is working, and that all of the conduits from the lgp tank to the pilot light are alright. The fact that the lpg tank is not empty again triggers a conversion to set the register for the lpg tank of the experiential system. Like the

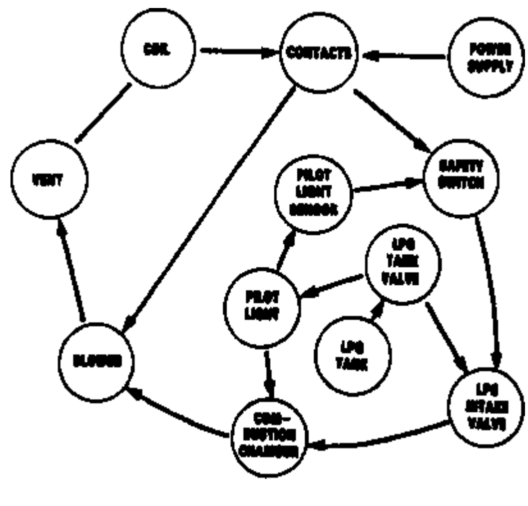

Figure 4. An Example Physical Knowledge Base for a Simple Heating System

previous conversion, it Is one-to-one and therefore straightforward. However, this Information proves to be useful to the experiential expert because the next question it would ask is if the 1pg tank is empty. Rather than asking the user an apparently redundant question, the experiential expert can acquire the information through the registers in the executor. In this way, the 1DM is capable of propagating knowledge acquired by one expert to the other expert, thus avoiding asking possibly dumb or repetitive questions.

Now suppose for this simple example, that the problem Is with the lpg intake valve being stuck shut. The experiential expert will therefore exhaust all of its possibilities, since It knows nothing of the lpg Intake valve, and the physical expert's knowledge base would appear as in Figure 5. Only the pilot light sensor, the safety switch, the lpg intake valve, the combustion chamber, and various conduits remain to be checked. These are parts of the system that the experiential expert knows nothing about.

The physical expert takes over diagnosis by systematically checking each of the nodes in the system to determine its input and output. it begins with the initial problem itself, the house being cold, which is represented by the conduit between the vent and the coll. It moves backwards from the vent to the blower and finally to the combustion chamber using the rules discussed above to guide the search. The physical expert knows that its output is not correct since the air coming out of the vents is not warm, so it looks for the source of the combustion chamber's input, which is the pilot light and the lpg intake valve. The pilot light is known to be alright but the lpg intake valve's state Is unknown. The system therefore asks the user to check the valve. This is found to be stuck shut. Once the valve Is loosened the furnace begins working properly.

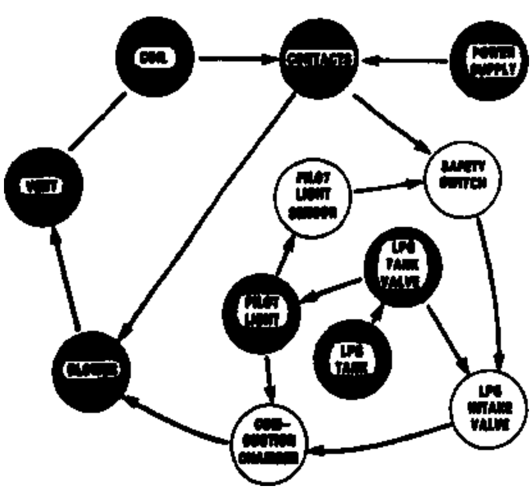

Figure 5. The Physical Knowledge Base at the Point where it Takes Control of the Diagnosis

In this last sequence, where the physical expert finally diagnoses the problem, we have demonstrated how a problem was found and solved that could not have been solved by the experiential expert because it lacked the proper knowledge. In such a simple example, of course, it would be possible to Include such knowledge in the experiential expert. However, in a larger system, such as the automotive electrical system that we are implementing, the inclusion of every detail of the wiring diagram would be very difficult and diagnosing from it would not be straightforward in the experiential expert. It knows about such things as the battery, the generator, the fan belt, and certain prominent wire connections, but more detailed information on the components and wiring is left to a more coherent and useful representation in the physical expert.

V RELATED WORK

Over the past several years much work has been done in addressing the problem of employing more fundamental knowledge in an expert system [1,2,3,5,7,8,12,13,15). However, most are concerned only with this deeper knowledge. Work by Patil et al. [12] is probably the most closely related to the work presented here because it is also concerned with integrating different levels of knowledge. In their work on ABEL, in the domain of diagnosing and treating electrolyte and acid-base disturbances, Patil et al. demonstrate how three different levels of knowledge representing the patient's state can be used to diagnose and treat the problem. It differs from our model not only in the number of levels of knowledge used but in how closely these levels are linked. In ABEL, the levels are designed to be compatible and consistent. The lower levels contain a more detailed version of the higher levels. Knowledge at one level can be passed consistently to another through a set of special functions. The IDM, on the other hand, does not necessarily require a consistent view of the problem between the experiential and the physical experts. Each expert is developed independently and integrated at completion. The experiential knowledge employed in the IDM resembles that used in the higher levels of ABEL. IDM's experiential knowledge is, in some cases, causal since the line between experiential and causal knowledge is somewhat indistinct. But the deeper knowledge employed in the IDM is more closely related to the device under diagnosis than is ABEL's. The physical expert of the IDM contains knowledge of the components of the device and how they behave and interact. Cause and effect is thus propagated via the physical and functional structure of the device rather than directly represented as in ABEL.

VI CONCLUSIONS AND FUTURE WORK

The IDM works well in the types of domains that it was designed for, namely mechanical/electronic diagnosis and repair. It even appears applicable to the medical domain, to debugging computer programs, and to failure modes and effects analysis of off-shore oil drilling platforms (using a different set of functional primitives). However, it does not appear applicable to such fields as mineral prospecting, speech recognition, and law. The experiential expert is most likely still viable, but the deep knowledge in these domains seems to require a different approach to representation than a set of functional primitives. The concept of processes [14] may be one possibility.

The IDM exhibits a way of providing a well-defined but flexible control over the propagation of knowledge between the different experts that erabody the diverse types of knowledge about a problem domain. Future work will include efforts to generalize the approach so that the conversion table in the executor can be automatically generated from the knowledge bases of the experiential and physical experts. Currently it is built by hand. We also plan to examine different top-level control strategies with respect to when to use each of the experts. This is an extremely difficult but important problem since it involves the question of how a human knows when to use each type of knowledge. Another area for research could be in identifying other types of deep knowledge that appropriately fit other domains such as the ones mentioned above. Finally, the use of several different kinds of deep knowledge that represent different aspects of the same device, such as electronic and mechanical, could be investigated.

REFERENCES

[1] Chandrasekaran, B., and S. Mittal, "Deep versus compiled knowledge approaches to diagnostic problem-solving," Int. J• Man"Machine Studies, 1983, pp. 425-436.

[2] Davis, R., "Reasoning from first principles in electronic troubleshooting," Int. J. Man-Machine Studies, pp. 403-423.

[3] deKleer, J., "The Origin and Resolution of Ambiguities in Causal Arguments," Proc. IJCAI-79 pp. 197-203.

[4] Fink, P,, J. Lusth, and J. Duran, "A General Expert System Design for Diagnostic Problem Solving," Proc* of IEEE Workshop on Principles of Knowledge-Based Systems, Denver CO, December, 1984.

[5] Genesereth, M., "The Use of Design Descriptions in Automated Diagnosis," Artificial Intelligence, Vol. 24, Nos, 1-3, Dec. 1984.

[6] Hart, P., "Directions for AI in the eighties," SIGART Newsletter, No. 79, January 1982.

[7] Kuipers, B., "Commonsense Reasoning about Causality: Deriving Behavior from Structure," Artificial Intelligence, Vol. 24, Nos. 1-3, Dec. 1984.

[8] Lusth, J., "Functional Modeling for Expanding Expert System Capabilities," working paper, 1984.

[9] Michie, D., "High-Road and Low-Road Programs," AI Magazine, Vol. 3, No.1, pp. 21-22, Winter 1981-1982.

[10] Minsky, M., "A Framework for Representing Knowledge," in The Psychology of Computer Vision, P. Winston (ed.), McGraw-Hill, Inc., pp. 211-277, 1975.

[11] Nau, D., "Expert Computer Systems," Computer, pp. 63-68, February 1983.

[12] Patil, R., R. Szolovits, and W. Schwartz, "Causal Understanding of Patient Illness in Medical Diagnosis," IJCAI-81, pp. 893-899.

[13] Rieger, C. and M. Grinberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms," Proc. IJCAI-77, pp. 250-256.

[14] Waterman, D., personal communication, December 1984.

[15] Weiss, S., C. Kulikowski, S. Amarel, and A. Safir, "A Model-Based Method for Computer-Aided Medical Decision-Making," Artificial Intelligence, Vol. 11, No. 2, pp. 145-172, 1978.