Pergamon

Expert Systems With Applications, Vol. 8, No. 1, pp. 67-76, 1995

0957-4174/95 $9.50 + .00

PRODE: A Shell for Industrial Diagnosis

GIUSEPPINA GINI AND PIERA SASSAROLI

AI and Robotics, Electronics and Information Department, Politecnico di Milano, Milano, Italy

Abstract—*PRODE (PROspective Diagnosis Expert), a shell for building expert systems based on prospective diagnosis, provides the domain expert with a knowledge acquisition interface and a suitable inference engine. Two kinds of knowledge contribute to modeling the diagnostic process. Failure knowledge expresses cause–effect relations between faults as well as the relations between the result of tests and the "degree of belief" about the presence of a fault. Strategic knowledge contains criteria to guide the selection of the appropriate action. A criterion refers to a particular viewpoint (such as the cost of a test or the estimated occurrence of a fault) and its application orders or prunes the set of the actions to take with respect to this viewpoint. The tool for knowledge acquisition provides the expert with a user-friendly interface in order to acquire both kinds of knowledge. Prospective diagnosis evolves acquiring new data to increase the knowledge about the system while the strategic criteria guide in this choice. The inference engine proceeds, alternating data acquisition and focusing, until the most probable cause of the malfunction is found and a repair is executed.*

1. INTRODUCTION

RAPID DIAGNOSIS of the machines in a plant is essential to meet production objectives. Repairs are often costly and late because people in the plant lack the knowledge or the experience needed to identify and fix the problem. Preventive maintenance plays an important role, but sudden malfunctions cannot be eliminated. An expert system can assist inexperienced people in this task. Because building expert systems is still a costly task, we approached the problem by defining a problem-oriented shell for diagnosis. The expert can directly enter the needed parameters into a shell that contains a knowledge representation and an inferential engine tailored to the problem. We have devoted our work to create a good interface between the expert and the shell.

A current trend is to provide knowledge acquisition tools to be directly used by the experts. For a review see (Nwana, Patou, Bench-Capon, & Shave, 1991). Usually the knowledge acquisition tool is strictly connected to the application domain, such as diagnosis (Diedrich, Ruhmann, & May, 1987; Eshelman, Ehret, McDermott, & Ming, 1987), evaluation (Klinker, Bentalila, Genetet, Grimes, & McDermott, 1987), and planning (Musen, Fagan, Combs, & Shortliffe, 1987).

Our application domain is diagnosis, in particular diagnosis of industrial equipment. A diagnostic problem has three components:

- the equipment that has malfunctioned;
- a set of tests that can be performed to better understand the state of the equipment;
- a set of repairs that can be prescribed to correct the malfunctions.

The goal of diagnosis is always to find the correct repair, possibly to find the correct cause.

Earlier KBS systems for diagnosis, notably MYCIN (Buchanan & Shortliffe, 1984), approach diagnosis as an empirical connection between some patterns of symptoms and some causes (infections), expressed by rules. No model of the system is used.

A diagnosis aimed at recognizing the current problem as an instance of a predefined hierarchical structure where the malfunctions are classified has been called *classificatory* diagnosis (Sticklen, Chandrasekaran, & Josephson, 1985).

When little information is available at the beginning of the diagnosis, the emphasis has been on choosing which data to acquire at a time. This has been called *sequential diagnosis* (Gorry & Barnett, 1968).

During diagnosis, to prove the presence or the absence of a malfunction, more data are needed. Tests provide such data. Some tests provide strong evidence of the presence or the absence of a malfunction, but at the same time they might present some drawbacks (for instance high cost or hazard). Others do not present these disadvantages, but provide less meaningful diagnostic data. *Prospective* diagnosis is a selection of tests based on their estimated effects and on the current state of the system under diagnosis (Cohen, Greenberg,

This research has been partially supported by Progetto Finalizzato Informatica, Obiettivo Qualitativi, of the CNR, and by ESPRIT, under the contract 2434. We thank Ing. Barni of Pirelli for his valuable time.

Requests for reprints should be sent to Giuseppina Gini, Dipartimento di Elettronicae Informazione, Politecnico di Milano, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy.

& DeLisio, 1987; Gruber & Cohen, 1987). We have chosen to approach diagnosis using this last paradigm.

Another important aspect is how to represent in the system equipment, malfunctions, tests, and repairs. The various trends so far emerged can be grouped into two main streams. One is representing the possible faults of the system, the other is representing the correctly functioning system.

Usually expert systems based on enumeration use a "shallow" knowledge representation. In the second direction, the pioneering work of (Davis, 1984; Genesereth, 1984) introduced the concept of "deep" knowledge. The intuition is that it is not necessary to know how a component part is broken but to prove whether its behavior is correct or not. Reiter (1987) developed a mathematical theory of diagnosis from first principles based on a finite set of faults, symptoms, and causal links. The computational intractability of this kind of deep knowledge is the main drawback of this approach. Moreover, the acquisition of the deep knowledge is itself a difficult task, and it is almost impossible for plants where the physical models are not well understood. Many authors like Reed, Stuck, & Moen (1988) and ourselves avoid reasoning from first principles and prefer reasoning on strategies.

Another aspect to consider is that knowledge contains a certain degree of uncertainty. This is due to the incompleteness of the knowledge base or to an ill definition of the problem or because the precise outcomes of actions cannot be predicted.

Whatever the cause, it is difficult to definitively assert chunks of knowledge and to reason with them in a two-valued logic. Two approaches to deal with uncertainty have emerged, the numerical and the symbolic one, well reviewed by D. A. Clark (1990).

Numerical approaches represent uncertainty by numerical estimates of it. This task may be very hard if not impossible in domains such as medicine and many other diagnostic tasks. For this reason we have chosen the symbolic representation with a 4-value scale to define the degree of belief. This approach offers user friendliness because it is based on a verbal scale of estimate.

In the following we present PRODE, a system for developing applications in the prospective approach. PRODE includes KA-PRODE, a tool for knowledge acquisition, as well as a suitable inference engine. We present our case study in Section 2. We describe the main features of PRODE in Section 3. Failure and strategic knowledge together with a description of how to acquire them are discussed in Section 4. In Section 5 we illustrate the organization of the knowledge base and the inference engine.

2. DIAGNOSIS AT THE CURING DEPARTMENT

Industrial maintenance aims at assuring the right level of operation of a plant. It involves planning the necessary revisions and tune-up of the machines.

When complex automation is employed, the maintenance becomes even more important because of the cost of the inactivity during failures. Even preventive and predictive maintenance cannot guarantee full operation of the plant.

At Pirelli the maintenance is organized in the fol-

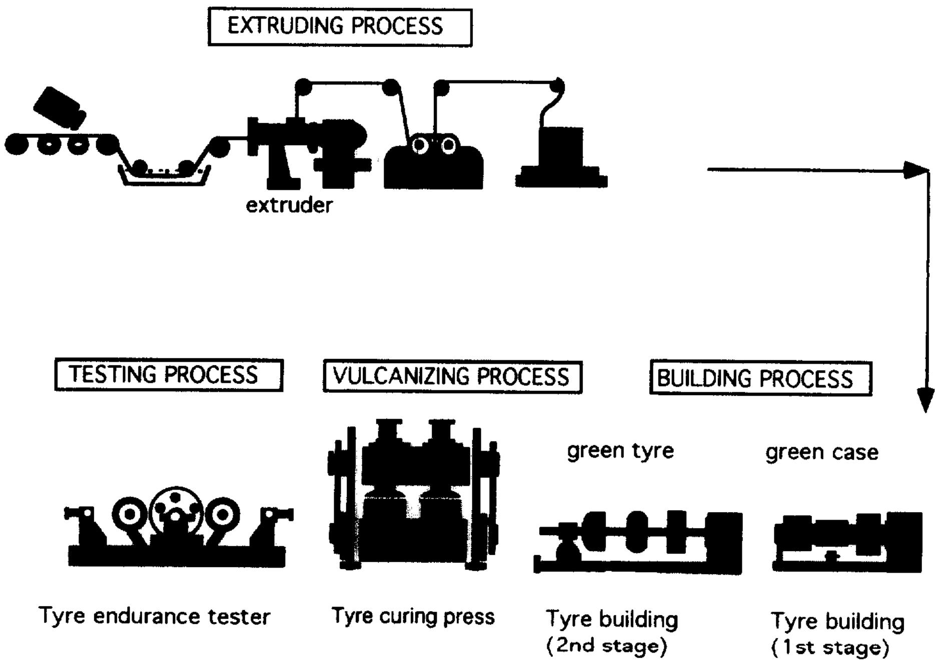

FIGURE 1. The last part of the tyre-building process.

TABLE 1
Phases of the Curing Department Cycle

1. Grasping the green tire
   1.1 the loader goes down
   1.2 the green tire is grasped
   1.3 the loader goes up with the tire
2. Opening the press
   2.1 unloading water
   2.2 the extractor–expeller device goes down
   2.3 the door starts opening
   2.4 pause
   2.5 the door is open
3. Unloading tires
   3.1 the extractor goes down
   3.2 the expeller goes up
4. Loading the tires on the mold press
   4.1 the loader goes down to the mold
   4.2 the tires are positioned and the loader goes up
   4.3 the curing rooms are positioned
5. Closing the machine
   5.1 the door starts closing
   5.2 pause
   5.3 the door closes
   5.4 the press works
   5.5 prepare for a new cycle

lowing services: *on failure maintenance:* in case of failures, experienced people arrive and make diagnosis, repair, and test of failures. If the experience is inadequate or the failure involves many parts, the central mechanical service is alerted; *preventive maintenance:* on a regular basis machines are tuned-up and consumables are filled; *predictive maintenance:* instrumentation is used by specialized people to check the vibrations of machines and to x-ray parts.

The central mechanical service has the function to make complex repairs, to make revisions and modifications and to send people to make repairs in the plant.

The need of expert systems is more evident in the first-aid activity, when no experienced people are available. The task is obtaining a guide in making diagnosis, so that less experienced people can be able to solve the failures.

The Pirelle plant for producing truck tyres is composed of stations for mixing, cutting, and preparing the green tyres. Then, as illustrated in Figure 1., the tyre enters the curing area, which constitutes the most crucial part of the process. The curing area is composed of many curing machines and is partially automated. Each curing machine is composed of a loader able to load 2 green tyres at a time, two molds where the tyres are cured, and a mechanical device to open the molds, transport the tires to a slide, and move the loader in position. They are coordinated by an electromechanical regulator. While the curing machine operates continuously, the repair team works in 3 shifts a day.

The activities of the curing cycle are listed in Table 1. We will discuss later the representation of the activities built in our system.

When the plant is working with mechanical automation or with PLC (as in the new lines) the diagnosis is a highly human-resources consuming task. Moreover it requires the presence of well trained people. The obvious desire of the factory management is to make the diagnostic knowledge available so that the errors could be individuated and the problems solved even in the presence of less experienced people and in a shorter time.

3. THE MAIN FEATURES OF PRODE

In prospective diagnosis knowledge about the state of the world is incomplete and the outcomes of actions cannot be predicted with certainty. The selection of actions is then based on their estimated effects and on the current state of the system under diagnosis. *Control knowledge* plays a crucial role in prospective diagnosis (Gruber & Cohen, 1987). Even though the final goal of the diagnosis is to find the fault, we reach that conclusion with many constraints: for instance with a limited budget, in a given time, and so on. As a consequence, in modeling a diagnostic process we distinguished two kinds of knowledge: failure and strategic.

Failure knowledge deals with the cause–effect relations among the faults, and with the local knowledge for every fault. Local knowledge relates the results of the available tests and the degree of belief about the presence of the fault they can provide.

Strategic knowledge is the knowledge that guides the inference engine in order to find the cause of the problem.

Traditionally, only the heuristic knowledge (which is a part of our failure knowledge) is acquired from the expert, while the whole control is embedded into the inference engine or, as in TEST (Kahn, Kepner, & Pepper, 1987), is expressed through procedures. PRODE provides a knowledge acquisition tool for acquiring both kinds of knowledge. Moreover, the range of the diagnostic applications that PRODE can effectively approach is greatly extended by its flexibility in also acquiring strategic knowledge.

During the acquisition of the failure knowledge, more strategies are available and are suggested by the system, firing a few heuristic rules.

Acquiring strategic knowledge means both defining control criteria and acquiring all the information used by these criteria, such as the estimated occurrence of faults, the type of the faults, the reliability of possible tests, and so on. This information is termed *control parameters.* A few of them are standard and predefined, while others can be provided by the expert. Control parameters might assume special or critical values, depending on exceptional situations arising during diagnosis. The expert can enter such situations and the corresponding values. If such situations arise during diagnosis, the control parameters get the new values and, as a result, can modify the hypothesis pursued by the inference engine.

Besides being a tool for acquiring knowledge, PRODE is a general architecture for expert systems based on prospective diagnosis. The problem-solving strategy is based on the execution of three steps:

- examining the failure knowledge and the control knowledge to find the right test to apply to individuate a cause of the malfunctions;
- applying the test and evaluating its effects on the current diagnostic hypothesis;
- applying control criteria to focus on the next hypothesis.

Humans are not used to making a diagnosis in a simple backward approach. Instead, the human expert is able to follow an hypothesis as far as a new test result suggests to consider a different possible fault. Similarly, the inference engine can "jump" to any hypothesis that becomes the most probable on the basis of tests, even if that hypothesis is not a possible cause of the current hypothesis.

4. KA-PRODE, THE KNOWLEDGE ACQUISITION TOOL

KA-PRODE (Knowledge Acquisition for PRODE) acquires both failure and strategic knowledge using a menu-driven dialogue. It asks for failure knowledge, for the values of the predefined control parameters, for new control parameters, and finally it generates the control criteria.

4.1. Acquiring Failure Knowledge

The system is modeled using phases, functions, component parts, and aggregate parts. Functions or phases describe the interactions between parts. A function is what a part is able to do. A phase is a temporal part into a process. Phases and functions are complementary aspects: many functions can be executed into a phase, or a function can be executed in more phases. A cyclic process can be easily represented as phases, as in our Pirelli case. On the other end, a functional description is useful in many diagnostic decisions. For this reason we can mix both phases and functions in our model.

Let *SF* (Set of Faults) be the whole set of faults of the system, and let *R* be the most general fault. For example, if the system is an engine, R could be "engine not OK." SF and the set of cause–effect relations among faults can be represented by a tree (the *failure-tree*), in which *R* is the root, the nodes are the faults, and the links in top-down direction denote cause–effect relations. Each link among A and B means that "B might cause A"; so the presence of A does not necessarily imply the presence of B.

We may enter, in any order, the faults of the system with their local knowledge. Each fault is represented by the couple "entity-name"-"malfunction". Local knowledge gives the relations between the results of tests and the degree of belief about the presence of the fault. In some cases a fault *F* is verified only if more causes $C1, C2, \ldots,$ and Cn are contemporarily verified. The conjunction of $C1, C2, \ldots Cn$ is called "multiple fault." Entity-names are maintained in a dictionary. In our Pirelli case, we have 50 objects in the dictionary.

The whole process of knowledge acquisition is menu-driven. At the beginning, the expert enters *R* and a few faults, in any order, so the initial tree is built. See in Table 2 some faults inserted bottom-up for the phase 1.1 of Table 1 in our Pirelli case:

The expert is guided to expand the tree through two elicitation strategies: breadth and first. When applying the first strategy, the expansion evolves by building one hierarchical level of the tree at a time, while in the second the expansion evolves by building one failure path (a chain of cause–effect relations from the leaf to the root) at a time. The failure-tree building process ends when no more faults and relations are entered.

For instance, a possible cause of "cam-box not in phase" can be "loader-descent not OK." The two strategies share the feature of walking through the tree either in "top-down" (from the root to the leaves) or "bottom-up" (from the leaves to the root) directions. In the bottom-up direction, KA-PRODE, starting from the lowest level faults, asks for new relations with upper level faults (effects) until the fault related to the root (final effect) is reached. In top-down direction, KA-PRODE, starting from the root, asks new causal relations with lower level faults (causes) until the most elementary repairable faults are obtained. Multiple faults can only be entered in top-down direction.

Another criterion can be applied at any moment: the "focus" criterion. KA-PRODE can focus on any cause–effect relation and ask to enter all the intermediate faults between the cause and the effect.

KA-PRODE uses a set of heuristic rules in order to propose the criterion, strategy, or direction to apply. These heuristic rules are based on practical and simple common sense ideas. For example, while starting the tree-building process, it suggests applying the path strategy in the bottom-up direction.

While building the failure-tree, two special conditions may occur, as illustrated in Figure 2. The new

TABLE 2
Some Faults of Phase 1.1: The Loader Goes Down to Grasp the Tire

cam-box not in phase
green tire in a bad position
sensor to detect green tire out of order
thermic problems in the loader engine
brake of loader engine out of order
relay of the loader engine out of order
loader engine burned

link *a* is a relation between the two faults, A and B, already connected by a causal link. If the new direct link is in the same direction as the previous one, this is a "short-cut." KA-PRODE keeps only the causal link that contains more information, that is, the one connecting more nodes. On the other hand, if the new causal link *b* is directed back from B to an old cause A, when we connect B and A again with a counter-directed link we obtain a "cycle." Since cycles may result in infinite loops, KA-PRODE eliminates them.

To describe *local knowledge*, let's first introduce the certainty-status of a fault. It represents the degree of belief about the presence of a fault. The range of values that the certainty-status might assume is a scale with four values: *excluded, probable, likely, certain.*

Whenever a new fault is entered, the expert is asked to specify a set of conditions for each value of the certainty-status. For example if the fault is "cam-box not in phase," we may say that this fault becomes likely when "operation-hours > 20".

A *condition* is a predicate, which assumes a truth value after an observation. An *observation* is a test, whose execution will provide the value to be used in order to assess the certainty status of the corresponding predicate. Direct human perception, instruments, or software packages may be needed to execute the tests. More than 50 observations are necessary in our Pirelli case.

An *observable entity* is a measurable attribute of a physical entity. It is represented as a frame with a value slot that will be filled with the value obtained from an observation. This value is supposed to be constant during diagnosis until a repair occurs. Every predicate refers to one or more observable entities.

An observation might also be a retrieval of data from historical data bases. In this case the expert selects the condition from a menu of predefined patterns of historical conditions. Suppose that "breakdown-number of ⟨object⟩ > ⟨number⟩" is the item, and that the expert fills it with "loader-engine" for ⟨object⟩ and "10" for ⟨number⟩; the condition "breakdown-number of loader-engine > 10" is obtained, and during diagnosis it will trigger, the execution of the appropriate retrieval action.

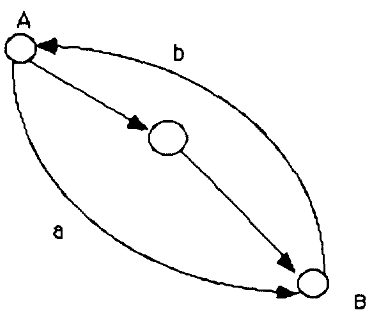

FIGURE 2. Short-cuts and cycles.

Each group of conditions referring to a particular value of the certainty-status is represented as a production rule (local rule) with the following structure:

if ⟨conditions⟩ then ⟨fault⟩ is ⟨certainty-status value⟩

where ⟨conditions⟩ denotes a conjunction of predicates, and ⟨certainty-status value⟩ denotes one value among "probable," "likely," "certain," or "excluded."

A production rule has the meaning of "the premise is evidence for the conclusion," so it can be considered as an evidential relation (Pearl, 1987) among the symptoms and the degree of belief about the presence of the fault.

4.2. Acquiring Strategic Knowledge

Strategic knowledge is knowledge about "what to do next." Strategic knowledge is a very crucial part of the expertise, because the criteria to consider might be numerous, heterogeneous, and of different priorities.

These criteria could be embedded in the inference engine. However, this solution is not acceptable because it becomes impossible to modify the criteria. The solution of providing predefined criteria stored in an appropriate knowledge base is better, but the expert cannot directly enter new specific control criteria. So, providing facilities for acquiring strategic knowledge becomes an important requirement of a knowledge acquisition tool.

The facilities provided by KA-PRODE for acquiring strategic knowledge (Mussi & Morpurgo, 1990) are suitable for the prospective diagnosis domain. As pointed out by Cohen et al. (1987), prospective diagnosis evolves in a context characterized by uncertainty and incompleteness: knowledge about the state of the system is incomplete, and the results of the actions required to achieve a goal are not known in advance. So we need to choose the "best" action on the basis of specific criteria.

For example, suppose that the cost of action A, which proves that F is certain, is higher than the cost of action B, which proves that F is probable. If the criterion "choose the cheapest action" holds, B is selected. Otherwise if the criterion "choose the action that proves the strongest diagnostic evidence" holds, A is selected.

4.2.1. *Control Parameters.* All control parameters are related to faults, observations, or repairs.

Some *standard* control parameters are usually present in any diagnostic application. They are predefined in PRODE, while the range of values that each parameter might assume is given by the expert.

1. The standard parameters related to *faults* are the *type* of the fault and its estimated *occurrence.* The type of a fault depends on the domain. For example,

in electromechanical domains the type could identify the fault as mechanical or electrical. The estimated occurrence is an a priori assessment that the fault occurs. For example, the estimated occurrence for the fault "engine fused" might be "low," while it might be "high" for "engine cold."

2. An *observation* is qualified by the standard control parameters *cost* and *reliability,* whose values represent an a priori assessment of the cost or reliability of the action needed to verify the corresponding condition.
3. When a new fault is entered, one or more *repair* prescriptions can be associated with it. Since terminal faults correspond to the most elementary repairable faults (leaves of the failure-tree), they are always associated with repairs. The repairs are qualified by the standard control parameters *cost* and *repair-time.* In our example, we have listed more than 40 repair actions, each associated with a time that can be set in each shift according to the competence of the repair team (usually the team is reduced at night).

The predefined range of values related to the standard control parameters can be extended. Moreover, the expert could ignore the request of a value for a control parameter, which maintains the default value "unknown".

The expert can extend the set of control parameters and tailor them to specific domain applications. For example, he/she might add a qualifier for the repairs, as *safety-hazard.* The new control parameters are asked after the failure-tree has been built because at that moment their need may become evident.

The expert can also define special or critical situations that, whenever they arise during the diagnosis, change the values of some control parameters. We call these situations "exceptional." An exceptional situation is defined through a pattern of conditions; their verification changes the defined value. For example, suppose that KA-PRODE asks the value of the new control parameter safety-hazard for the repair "clean-up the store," and that the answer is "low." Then KA-PRODE asks for possible exceptional situations, and the expert answers "poisonous gases are stored." The new value of safety-hazard in this case can be "high." In any diagnosis, if the exceptional situation arises, the value "high" will be used instead of "low."

4.2.2. *Control Criteria.* The control criteria used in PRODE are classified into three groups depending on their role:

1. *Hypothesis selection:* selecting from a set of possible faults (hypotheses), the one to verify. Criteria for hypothesis selection are used by the inference engine to focus on a fault, as in the following example. Let the value of the control parameter "estimated occurrence" be in the set (low, medium, high). Suppose that the certainty-status of the fault "engine surges" is "certain," and that both "engine fused" (whose estimated occurrence is "low") and "engine cold" (whose estimated occurrence is "high") are candidate hypotheses. If the criterion "select the faults with highest value of estimated occurrence" holds, the inference engine selects "engine cold" as the hypothesis to pursue.

All the criteria for hypothesis selection are instances of the following general pattern:

select the faults with $\langle$*limit*$\rangle$ *value of* $\langle$*control parameter*$\rangle$

2. *Observation selection:* identifying, for a given fault, both the "best" value of certainty-status to be verified and the appropriate local rule to conclude it. Criteria for observation selection are "local" to a fault, and "best" depends on the selected criterion. To reach a diagnosis, PRODE tries to verify the presence of a fault with the maximum degree of certainty ("excluded" or "certain") compatible with a given criterion, for instance, the minimum cost of the observations.

The criteria for observation selection are instances of the pattern:

minimize/maximize$\rangle$ *the* $\langle$*control parameter*$\rangle$ *of observations*

3. *Repair selection:* selecting, from the set of repairs related to a fault, the most appropriate one.

The criteria for repair selection are instances of:

select repair actions with $\langle$*limit*$\rangle$ *value of* $\langle$*control parameter*$\rangle$

where $\langle$limit$\rangle$ denotes "highest" or "lowest," and $\langle$control parameter$\rangle$ denotes any control parameter of the related faults, observations, or repairs. These criteria are generated automatically and stored as metarules on the basis of the acquired control parameters. For the nonstandard control parameters, the expert is asked to provide the values of $\langle$limit$\rangle$.

Let us give an example. The criterion for observation selection is C1) "minimize the cost of observations." Let the cost (with values: low, medium, high) be a control parameter for repairs. Consider now the fault F: the cost of its associated repair is "low," the costs of the observations are "low" to prove that F is "likely," and "high" to prove that F is "certain" or "excluded". The metarules M1 and M2 are generated.

M1) "if cost of the best repair $\leq$ low and
cost of the best rule proving F likely <
cost of the best rule proving F certain and
cost of the best rule proving F likely <
cost of the best rule proving F excluded
then try the best rule proving F likely"

The meaning of "best rule" is related to the given criterion; since C1 holds, among the rules concluding the same value of certainty-status, the one with the minimum cost of the observations needed to verify the premise is selected.

M2) "if cost of the best rule proving F certain <
cost of the best rule proving F excluded
then try the best rule proving F certain"

During diagnosis, if no repair is associated with F or if the cost of the best repair is "unknown," PRODE cannot apply M1; so M2 might be selected.

The term "best" for repairs means that, whenever more repairs are associated with a fault, one is selected on the basis of the control criteria for repair selection.

A built-in *meta-criterion* selects the most specific metarule among the fireable ones (in this example, M1 would be selected).

The selection among faults is executed in the following two steps: (a) criteria for observation selection are applied to each fault and a local rule for each one is obtained; (b) the fault whose local rule is the "best" is selected.

The expert defines for each group a priority in the criteria. Whenever the values of the control parameters of the selected criterion are unknown, and the criterion cannot be applied, the next will be selected.

5. ARCHITECTURE OF PRODE

We present now the knowledge representation schema and the inference engine that make up an important part of PRODE.

To describe the internal organization of knowledge, we will refer to Figure 3. A fault is represented by a frame with a variable number of slots. The fixed slots contain the causal relations (list-causes, list-effects), the repairs (list-repairs), and the certainty-status; other slots are related to the control parameters of the faults (estimated-occurrence, type, . . .) and contain the values provided during the knowledge acquisition phase.

A demon, corresponding to an exceptional situation, might be associated with a slot. Whenever a demon is activated, the value of the corresponding slot is changed.

A set of local rules might be associated with the certainty-status slot. Each observable entity appearing in the condition of a local rule is represented by a frame. The action slot, appearing in an observable entity frame, refers to the observation needed to provide a value for the observable entity.

Observation and repairs are also represented by a frame. The slots named text, appearing in both repair and observation frames, contain a description of the related action. The values of the slots are provided during knowledge acquisition and might be changed during

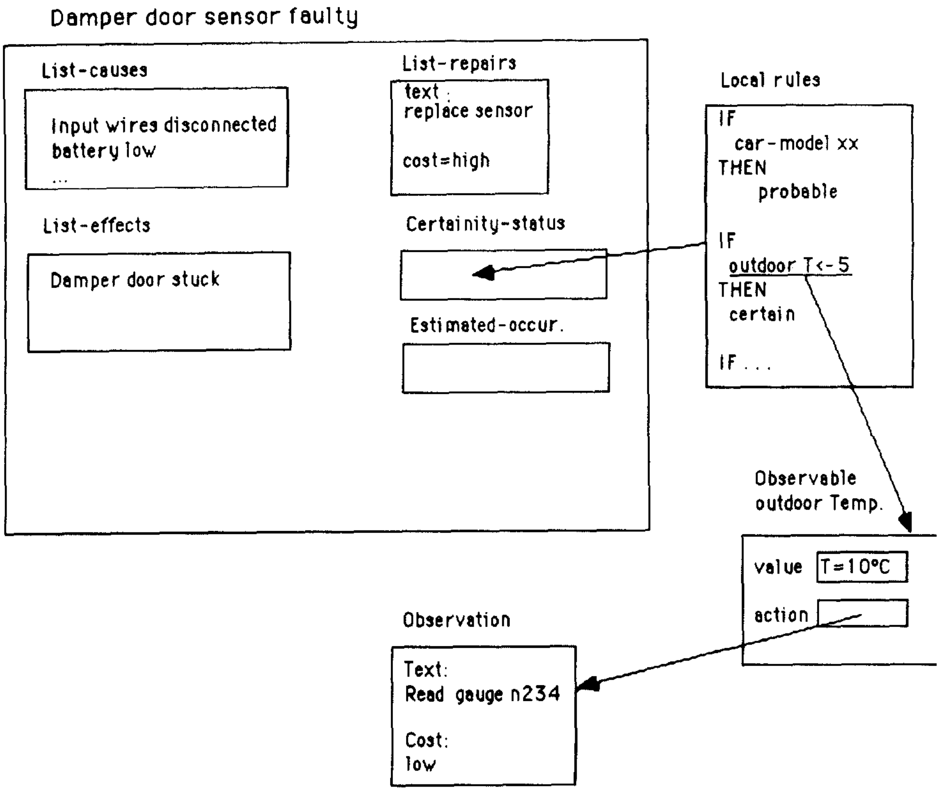

FIGURE 3. Knowledge organization.

the diagnostic process if an associated demon is activated.

From the criteria for hypothesis selection, one of the possible causes of the fault is selected; from the criteria for observation selection, one of the local rules of the fault is selected; and from the criteria for repair selection, one repair is chosen.

Let us look at the inference engine. The aim of the inference engine is to prove the presence or the absence of a fault with the maximum degree of certainty and detail. The maximum degree of certainty corresponds to the values "certain" and "excluded" of the certainty-status, while the most "detailed" faults are those related to the leaves of the failure-tree. The inference engine activity is essentially based on the iterative execution of three phases, as illustrated in Figure 4:

1. formulate hypotheses from the acquired information;
2. focus on an hypothesis (fault) to be pursued;
3. acquire new data to verify this hypothesis (presence of the fault).

Whenever new information is acquired, the inference engine tries to verify all of the local rules containing this information in their premise and creates a list of hypotheses. When the list is completed, the inference engine identifies (focus) an hypothesis and asks for new information in order to verify it. The new information fires local rules in order to formulate new hypotheses. The diagnostic process works through the following steps:

1. Starting from the initial situation, in which the certainty-status of all the faults is "unknown," the user provides the available symptoms. As a consequence, some local rules fire, and some faults might get a new certainty-status. The value "certain" is assigned to the certainty-status of the root R.
2. Let C be the set of faults whose certainty-status is "certain" (C is never empty because it contains at least R). A pruning of C is obtained in the following way. For each path starting from R and containing at least one "certain" fault, all faults except the most "specific" one (i.e., the farthest from R) are deleted from C. Let LC be the subset of the elements of C that are leaves, and NLC the subset of elements of C that are not leaves. The "best" fault R' to focus on is chosen in LC, if not empty, otherwise in NLC.
3. R' becomes the root of the new subtree to be examined. Since in this subtree the only "certain" fault is R', we consider the set P of faults of this subtree whose certainty-status is probable. If P is not empty, a pruning of P is carried out, applying the method described in ii. Otherwise, a fault whose certainty-status is "unknown" is selected into the subtree rooted in R', applying the control criteria for hypothesis selection. Let F be the selected fault.
4. The attention is now focused on F. To "improve"

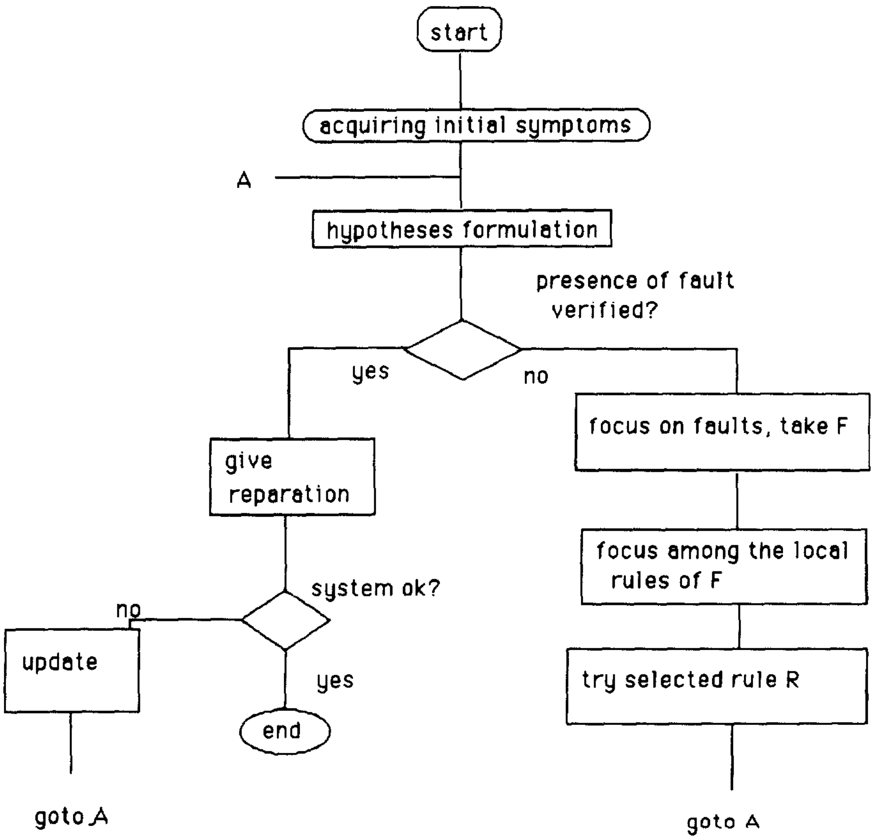

FIGURE 4. A simplified view of the inference engine.

its certainty-status the "best" local rule is selected by the control criteria for observation selection. Namely, if F is "probable," a local rule concluding F "certain" or F "excluded" is selected, while if F is "unknown" a local rule concluding any certainty status is selected. To fire the selected rule, new information (the results of the observations in the premise of the rule) is asked for. So new hypotheses are formulated, and the process resumes from point ii. When a repair action is successful the inference engine stops.

PRODE provides a repair prescription whenever the certainty-status of a fault (a leaf in the failure-tree) is different from excluded. In step ii, if an element of C is a "certain" leaf, a repair prescription is given and the process might end. Moreover, if in step iii, an element of P is a leaf and its best value is "likely," a repair prescription is given for this fault, and the process might end. After the repair, if the problem is still present, the possible changes in the system under diagnosis are shown. Then diagnosis restarts from step i after assigning "unknown" to the certainty-status of all the faults.

The inference engine has been designed to simulate the behavior of a human expert. The expert might find an observation result that "triggers" in his/her mind a possible fault not belonging to the set of candidate hypotheses. This might happen in the inference engine activity if at least one rule fires after an implication finding. In this case a new fault might become "certain," and it could be taken into account even if it was outside of the subtree under examination. This cannot be handled by a simple backward-chaining. Moreover, our metaknowledge is different than in traditional rule-based systems. Metaknowledge is usually represented by metarules whose effect is pruning the tree or selecting into the set of possible direct causes of a fault (Davis, 1980). In our approach, metaknowledge corresponds to control criteria for hypothesis and observation selection, and their effects, as described above, are more powerful than those provided by traditional metarules.

6. CONCLUSIONS AND FUTURE WORK

PRODE, a system for solving problems in prospective diagnosis domains has been presented. It provides a knowledge acquisition tool, a knowledge base representation of the acquired knowledge, and an inference engine.

The design of PRODE has been motivated by a previous experience in building the diagnostic system for the curing press department of the Pirelli plant. We have discovered that the use of shallow knowledge was appropriate, but that experts used many strategies in searching for faulty parts. We have completed the design and the implementation of PRODE, that is written in Common Lisp on a TI Explorer.

The choice of diagnosis based on user-defined strategies has some important advantages over simple rule-based systems and over diagnosis from first principles. For common and known faults, the system may perform as well as a shallow system. The causal model is not needed, and indeed it is not available in most of the diagnostic applications; any causal connection between parts in the system can however be represented through links and local rules.

PRODE is a prototype expert system motivated by the need of a Pirelli curing area. After appropriate testing of the completeness of the knowledge base, we will work to assess the system in the field. The main drawback of this prototype is the lack of information from the site and the control room. The expert system should be installed on a computer in the curing area, because all of the observations should require some human intervention. The next step will be replacing the controllers with PLC, and installing a network in the plant. In this case the expert system can be installed in the control room, where all the sensor readings will be available, and maintenance people will be alerted only to make repairs.

As an expert system, PRODE still presents some limits that we intend to overcome in future research. The main area of investigation is the acquisition of control criteria. Even though the dialogue between KA-PRODE and the expert is simple, the mental effort required to answer some questions about defining new control parameters, defining and ordering control criteria, etc. might be heavy. We are planning to add a suitable module to elicit strategic knowledge asking the expert about a few specific examples. On the basis of the information acquired in this dialogue, the general criteria should be generated applying inductive learning techniques.

REFERENCES

Buchanan, B. & Shortliffe, E. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison Wesley.

Clark, A.D. (1990). Uncertainity management in AI. *Artificial Intelligence Review,* **4,** 109–146.

Cohen, P.R., Greenberg, M., & DeLisio, J. (1987). MU: A development environment for prospective reasoning systems. In *Proceedings of AAAI,* Seattle, WA.

Davis, R. (1980). Metarules: Reasoning about control. *Artificial Intelligence,* **15,** 79–222.

Davis, R. (1984). Diagnostic reasoning based on structure and behavior. *Artificial Intelligence,* **24,** 347–410.

Diedrich, J., Ruhmann, I., & May, M. (1987). KRITON: A knowledge-acquisition tool for expert systems. *Int. J. of Man–Machine Studies,* **26,** 29–40.

Eshelman, L., Ehret, D., McDermott, J., & Ming Tan. (1987). MOLE: A tenacious knowledge acquisition tool. *International Journal of Man–Machine Studies,* **26,** 41–54.

Genesereth, M. (1984). The use of design descriptions in automated diagnosis. *Artificial Intelligence,* **24,** 411–436.

Gorry, G.A., & Barnett, G.O. (1968). Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, **1**, 490–507.

Gruber, T. (1987). Acquiring strategic knowledge from experts. *Proceedings of the 2nd AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

Gruber, T., & Cohen, P.R. (1987). Design for acquisition: Principles of knowledge-system design to facilitate knowledge acquisition. *International Journal of Man–Machine Studies*, **26**, 143, 159,.

Kahn, G., Kepner, A., & Pepper, J. (1987). TEST: A model-driven application shell. Proceedings of the AAAI, Seattle, Washington.

Klinker, G., Bentolila, J., Genetet, S., Grimes, M., & McDermott, J. (1987). KNACK—Report-driven knowledge acquisition. *International Journal of Man–Machine Studies*, **26**, 65–79.

Musen, M.A., Fagan, L.M., Combs, D.M., & Shortliffe, E.H. (1987). Use of a domain model to drive an interactive knowledge-editing tool. *International Journal of Man–Machine Studies*, **26**, 105–121.

Mussi, S., Morpurgo, R. (1990, August). Acquiring and representing strategic knowledge in prospective diagnosis domain. *Expert Systems*, **7**(3).

Nwana, H.S., Paton, R.C., Bench-Capon, T.J.M., & Shave, M.J.R. (1991). Facilitating the development of knowledge based systems, a critical review of acquisition tools and techniques. AICOM, **4**, N 2/3.

Pearl, J. (1987). Embracing causality in "formal reasoning." *Proceedings of the AAAI* 87, San Mateo, CA: Morgan Kaufmann.

Reed, N.E., Stuck, E.R., & Moen, J.B. (1988). Specialized strategies: An alternative to first principles in diagnostic problem solving. *Proceedings of AAAI* 88, pp. 364–368. CA: Morgan Kaufmann.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, **32**, 57–95,.

Sticklen, J., Chandrasekaran, B., Josephson, J.R. (1985). Control issues in classificatory diagnosis. *IJCAI* 85, CA: Morgan Kaufmann.