The Knowledge Representation and the Inference Strategy for Machine Diagnostic Expert System

주 석 진*
권 영 식**

Abstract

This paper describes an artificial intelligence approach to machine diagnosis. Firstly, it considers how the knowledge could be organized and represented. Secondly, it considers which inference strategy could be chosen for contingent situations for the purpose of rationality, efficiency and user-friendliness. Finally, the prototype based on the suggested model is introduced briefly.

요 지

본 논문은 인공지능기법을 이용한 기계고장진단 전문가시스템에서의 적절한 지식표현과 추론방식을 개발하는 것을 목적으로 한다. 끝으로 제안된 모형에 의한 전문가시스템 원형이 제시된다.

1. Introduction

The problem in machine diagnosis is to isolate in the most efficient manner, the cause for a particular failure andrecommend a repair. Machine diagnostic systems that can automatically pinpoint failed components and recommend a repair procedure have been a long standing theme of AI research.

The reasons why machine diagnosis problems are investigated by many AI researchers as follows :

• Machine down time is very expensive in some manufacturing areas. For instance, one hour of machine down time could cost $11.000 in semiconductor manufacturing.

• Domain experts who can tackle complex machine diagnosis problems well are very scarce and their time is very expensive.

• Conventional documentation for troubleshooting is seldomly used by field mechanics, because it is often too massive and difficult to find the right page for the specific problem at hand.

• An on-boardtroubleshooting approach has many problems, such as inflexibility, incomplete domain knowledge, crypticoutput, and lack of an explanation capability.

Expert systems, by definition, capture human expertise into computer programs and use the knowledge gathered to deal with complex, and unstructured problems[6], [15]. For problem areas such as machine diagnosis, AI or more specifically expert system techniques can be applied to realize improved performance if the systems are implemented to resemble a human expert's reasoning processes.

In general, there have been two approaches in diagnostic problems[8]. One approach is based on the status of operation of observed systems after delineating the function of parts of the system and the relationship among them[4]. It is also called causal reasoning or reasoning from first principle[12]. In the other approach, like in MYCIN[2], the empirical knowledge on the domain is mainly emphasized.

The model which will be explained is based on the weak causal reasoning consisting failure mode for solving diagnostics problems and applied to building the prototype of our diagnostic expert system.

*경기대학교 경상대학 경영정보학과
**동국대학교 공과대학 산업공학과
접수 : 1989. 4. 24.

2. Diagnosis Model Based on Hierarchical Failure Mode

All human diagnosticians, whether they work in machine repair or medicine have certain characteristics in common. Both groups have and internal mental model for the domain. This model is a body of knowledge about the parts of the mechanism or organism they are trying to fix and about how those parts fit together.

This model is closely tied to two additional knowledge sources : the expert's formal understanding of the laws of the domain(such as electrical theory), and a large loosely structured body of knowledge consisting of his common sense and experience gained by simply living in the world. Taken together, these three knowledge sources are very powerful and enable human beings to solve new problems by reasoning them through. This process is called "causal reasoning" or "reasoning from first principles"[12].

However, we discovered an interesting fact about causal reasoning : experts rarely solve problems this way. They only perform causal reasoning when absolutely necessary, such as when confronted with a new situation. Most of the time experts work from a different representation, a much simpler model that is derived from the original causal model but is much easier to work with.

This second model which can be called as weak causal model consists of failure modes and their relationships to each other, and represents the diagnostic problem in terms of which low-level, repairable failures cause the higher-level, observable failures(Figure 1). It retains most of the diagnostic power of the original causal model, but does not require such difficult reasoning.

This model seems to be a natural and proper method for capturing a human expert's knowledge, since the knowledge that experts exhibit seems most naturally represented as a hierarchical structure of failure modes built up from years of experience.

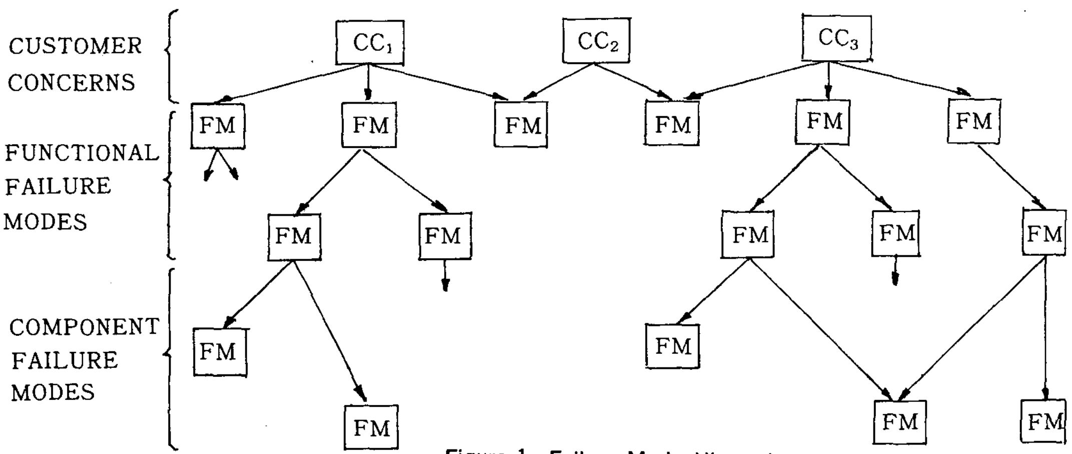

Figure 1. Failure Mode Hierarchy

In this model, the failure modes are linked "downwards" to the more specific failures which are possible causes for them, and "upwards" to the more general failures which they can cause by the due-to relation and its inverse, lead-to. This basic failure mode hierarchy is augmented with all the supporting knowledge used by experienced technicians.

This knowledge includes test procedures used to confirm or reject failure modes : repair procedures used when a failure mode is confirmed : and exception condition describing how the diagnostic strategy should be modified depending on certain conditions. By navigating in this failure mode hierarchy, we can reach the component failure responsible for the customer concern.

3. Knowledge Base Organization

The knowledge base consists primarily of a hierarchy of failure mode. At the top level of the hierarchical structure is a customer concern which is itself a failure mode. The customer concern is linked to one or more middle-level, functional failures that are possible causes for it.

These functional failures in turn lead to bottom-level, component failures that are repairable. The causal relationship between any tow failure modes is established by the DUE-TO relation and its inverse(Figure 2).

A special type of failure mode called a failure mode class(fmc) is sometimes used for knowledge base modularization and for providing better explanations to the user. A failure mode class is different from a simple failure mode in that it groups together related failure modes under a single classification.

Thus a failure mode class is confirmed if any one of the failure modes underlying it is confirmed while a simple failure mode is confirmed or disconfirmed as a result of a test that is attached to it by the HAS-TESTS relation.

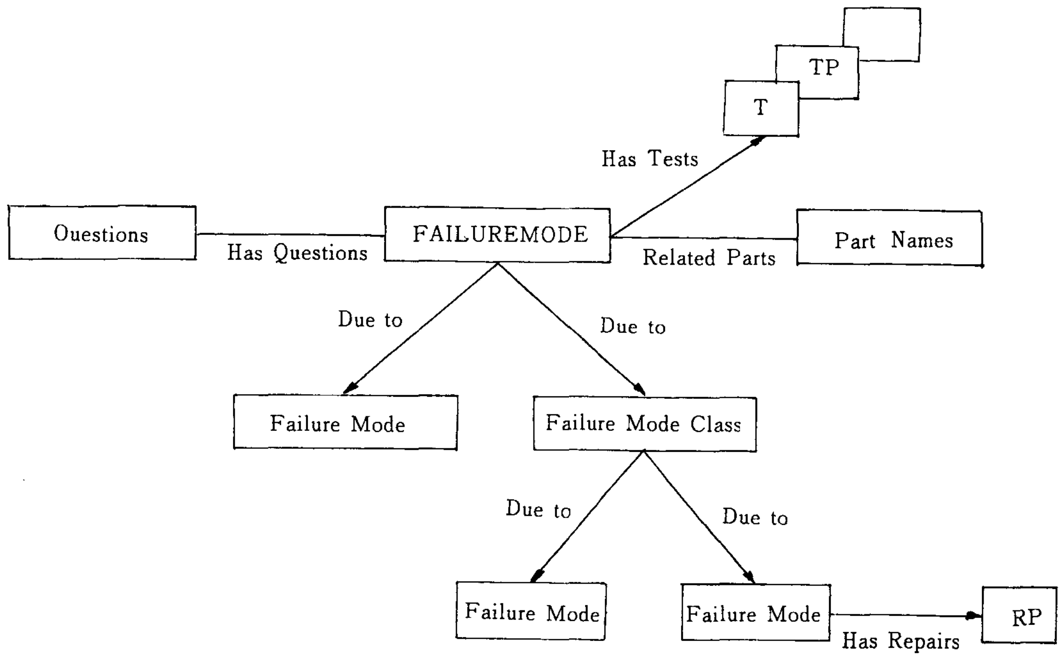

Figure 2. Failure-Modes and Relations

The knowledge base is arranged in such a way that the failure mode are investigated in the order of likelihood and cost of performing the related tests. However, in the course of a troubleshooting session, the knowledge base may be modified to reflect the changingstate of the machine by removing or shuffling some of the chil failure modes. Also, the knowledge base is tailored to the type of the machine under diagnosis.

Often, additional information is required from the user before investigating further a confirmed failure mode. If additional information is needed then some questions are attached to the failure mode by HAS-QUESTIONS relation. The answers to such questions will normally cause a set of meta rules to fire, where by the knowledge base is dynamically altered.(See Section 6.)

4. Knowledge Representation

Rule, semantic nets and frame are alternative methods for representing knowledge.

The concept(failure mode in this case) at each node in the knowledge base might be represented well if it is defined by a collection of attributes and the values of those attributes. Moreover, it would be desirable if the concepts are organized in a hierarchical structure with the specific relations. Frame is appropriate for these purposes. The knowledge representation of failure mode and it related objects by frame is a follows.

The prototype failure mode frame appears as follows :

```
{FAILUREMODE
        DESCRIPTION :
        DUE-TO :
        DUE-TOS-ARE-EXHAUSTIVE : yes
        HAS-TESTS :
        HAS-REPAIRS :
        HAS-QUESTION :
        RELATED-PARTS :
        OCCURS :  unknown}
```

The prototype frame is defined by the frame representing the typical member of a set of entities. The prototype failure mode frame defines the attributes and characteristics typical of all failure modes.

The slot definitions for the failure modes in the knowledge base are as follows :

- DESCRIPTION-a string used for explanation.
- DUE-TO-the failure modes which this failure mode is due to. Ordered by the expert according to the order in which they should be investigated based on likelihood and cost.
- DUE-TOS-ARE-EXHAUSTIVE-YES if the failure modes listed inthe DUE-TO slot is complete;No otherwise. Used by the interpreter to determine if the "rule-out diagnostic strategy" is appropriate.
- HAS-TESTS-the tests which can be used to confirm or disconfirm this failure mode.
- HAS-REPAIRS-the repairs to fix this failure mode(normally a component failure).
- HAS-QUESTIONS-the questions to be asked. The answers will alter the knowledge base before any tests are performed.
- RELATED-PARTS-parts related to this failure mode. Used in graphics display.
- OCCURS-YES if this failure mode is confirmed(known to occur);No if it is disconfirmed(known not to occur);UNKNOWN otherwise.

This basic structure of failure modes is enriched by other supporting knowledge. Associated with each failure mode is one or more tests as specified in the HAS-TESTS slot, which provides evidence to confirm or disconfirm the failure mode.

Meta-value information is attached to individual tests inthe HAS-TESTS slot. This information indicates what value(s) of the possible test results should be considered as a confirming result for that failuremode. The prototype frame for representing tests is shown below with an explanation of the slot values :

```
{{TEST
        DESCRIPTION :
        PROMPT :
        POSSIBLE-RESULTS :
        RESULT :            }}
```

- INSTANCE-must be TEST.
- DESCRIPTION-a string used for explanation.
- PROMPT-a string which is displayed to the user, instructing him/her to perform the test.
- POSSIBLE-RESULTS-all the possible result values the test can yield.
- RESULT-the actual result of the test, if any.

The HAS-TESTS slot in the failure mode frame is disjunctive, that is for failure modes which have more than one test listed in the HAS-TESTS slot, one confirming test is sufficient to confirm the failure mode. Where several tests are reguired to confirm a failure mode (the conjunctive case), a new type of frame is introduced in addition to the test frame called a test procedure.

The test procedure name rather than a test name is then put into the HAS-TESTS slot of the failure mode. The test procedure frame then contains a slot called HAS-STEPS where individual test names are specified. All tests listed in the HAS-STEPS slot must beconfirmed before the test procedure is confirmed.

The prototype test-procedure frame isshown below with a description of the slot values following :

||TEST-PROCEDURE

HAS-STEPS : ||

• HAS-STEPS-tests that comprise the test procedure.

For each failure mode at the component level(corresponding to a leaf node of the hierarchical structure), a repair procedure to rectify the fault is specified in the HAS-REPAIRS slot. The prototype frame for repair is shown below as well as a description of the slot values :

||REPAIR

DESCRIPTION :

HAS-PROCEDURE : ||

• DESCRIPTION-a string used for explanation.

• HAS-PROCEDURE-a string which is displayed to the user, instructing how to perform the repair.

5. Inference Strategies

5.1 Backward Chaining Inference in the Failure Mode Hierarchy

The reasoning method that forms the backbone of inference mechanism is backward chaining or goal-driven reasoning. A diagnosis starts with a top-level failure mode which is chosen according to customer complaints.

The system (Prototype of application example system called DESACS is described in section 7.) looks at the DUE-TO list of this selected customer concern failure mode, and it considers the first element of the DUE-TO slot to be the most likely cause.

Now the system tries to find the evidence for this new failure mode using test(or test procedure) which is listed in HAS-TESTS slot.

If the test result is same as the meta-information attached to the test name, that failure mode will be confirmed. Other wise the system move down to the next cause for the customer concern. The first confirmed failure mode is regarded as the cause of parent failure modes. The cycle then repeats, so the system continually going deeper and deeper into the DUE-TO hierarchy, until it confirms a component failure mode which has no child failure modes but has a recommendation(repair procedure).

5.2 Changing the Focus of Concern by Volunteered Information

The normal user follows the systems's instructions blindly in the consultation process, and is able to arrive at a correct diagnosis. But a highly skilled user wants to use the system as an expert advisor and, if he wishes, provide voluntarly additional information that the system uses to arrive at a solution faster.

For these volunteered informations, the system has an additional forward chaining capability. During any point in the diagnostic process, the user is allowed to assert new-found facts about any failure mode or test. Then the system changes its focus of concers to the new point in the failure mode hierarchy and continue reasoning from there.

5.3 Intelligent Behavior by Best Guess.

Where all the failure mode, the system automatically concludes that the last item is the cause. This rule-out strategy makes the system more smart, because it eliminates the request for redundant tests.

But this strategy cannot be used when the DUE-TO list are not exhaustive. The exhaustive DUE-TO list is used for another example of intelligent behavior. When theuser is asked to perform a certain test, he/she cannot or will not to do it by replying UNKNOWN. In this case the system goes on without the desired information. In the long run, if only one failure modeis remained as UNKNOWN state and others are all rejected, then system concludes that the unknown failure mode is the cause of its parent.

6. Dynamic Modification of Knowledge Base using Meta Rules

Asit mentioned before, the knowledge base is arranged in such a way that the failure modes are investigated in the order of likelihood and cost of performing the related tests. But it only describes the ordinary ways, and just like human expert, the system needs a way to modify its behavior when confronted with exceptions and unusual conditions.

To replicate this aspect of problem solving, the exception rules are attached to the failure modes. These meta rules which are invoked as their failure modes are confirmed modify the knowledge base to handle special conditions.

For example, a rule might say that if the engine over heat problem to a car only occurs with the air-conditioner running, then there is a higher like lihood that improperly installed air-conditioner will cause that problem. The knowledge base is modified by removing or rearranging some of the failure modes in the DUE-TO list. This modified knowledge base can thus be used for finding another cause in multiple diagnoses.

7. Application Example

The DESACS, an prototype expert system which can diagnose and explain the possible cause of failures in the cooling system of an automobile, was developed as the result of our effort to apply the previously described model.

The system is comprised of four major components, the knowledge base, the interpreter, the explanation facility and the user interface and implemented using knowledge Craft on TI Explorer[17].

The knowledge base of DESACS contains 64 failure modes including three cooling system related customer concerns : Engine overheat, False indication of engine overheat, and Engine failing toreach normal operating temperature. A part of failure hierarchy in knowledge base is shown in Figure 3.

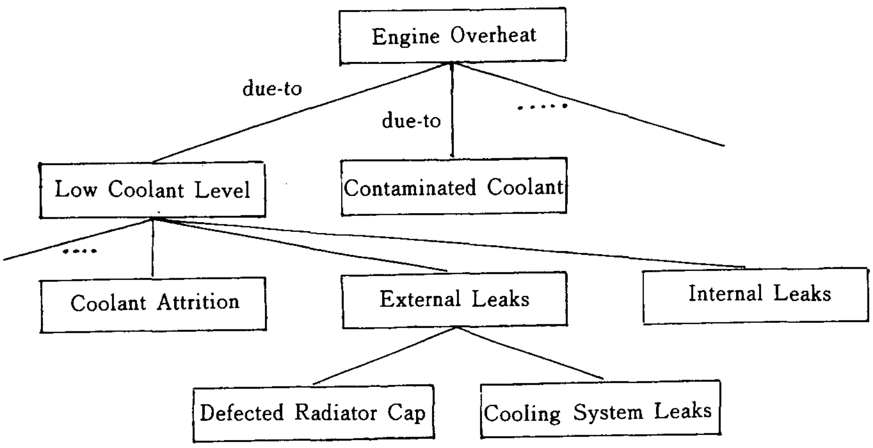

Figure 3. A part of DESACS knowledge Base

In the folliwing example frame, the functional failure mode Low coolant Level is shown to be one of possible causes of customer concern Engine Overheat[1].

Defected Radiator Cap, a component failure mode, could cause External Leaks which, in turn, is one of possible causes of Low Coolant Level[2, 3, 4].

Low coolant level could be proved by visually examining coolant level[5], whereas Radiator Pressure Cap Test using Rotunda 21-012 Radiator Pressure Tester should be performed for checking radiator cap[6].

The defected radiator cap must be replaced according to repair procedure specified in rp-replace-radiator cap frame[7].

```
(1) || FM-ENGINE-OVERHEAT
       INSTANCE : customer-concern
       DUE-TO : fm-low-coolant-level
                fm-contaminated-cooland
                fm-fan-problem
                fm-air-flow-obstruction
                fm-lack-of-coolant-flow
                fm-low-oil-level
                fm-improperly-installed-air-conditioner
                fm-excessive-loads
                fm-other-engine-related-problems
       DESCRIPTION : "Engine Overheat"
       HAS-TESTS : t-verify-engine-overheat
                   (confirming-results YES)
       HAS-QUESTIONS :
           q-engine-overheat-when-air-conditioner-running
           q-engine-overheat-under-severe-driving-condition ||
(2) |  FM-LOW-COOLANT-LEVEL
           INSTANCE : FAILURE-MODE
           DESCRIPTION : "Low coolant level"
           DUE-TO : FM-OBVIOUS-EXTERNAL-LEAKS
           FM-HEATER-CORE-LEAKS
           FM-OBVIOUS-INTERNAL-LEAKS
           FM-COOLANT-ATTRITION
           FM-EXTERNAL-LEAKS
           FM-INTERNAL-LEAKS
       DUE-TOS-ARE-EXHAUSTIVE : no
       HAS-TESTS T-VISUAL-INSPECT-COOLANT-LEVEL
                 (confirming-results Low)||
(3) || FM-EXTERNAL-LEAKS
       INSTANCE : fmc
       DESCRIPTION : "External leaks"
       DUE-TO : FM-DEFECT-RADIATOR-CAP
                FM-COOLING-SYSTEM-LEAKS
       DUE-TOS-ARE-EXHAUSTIVE : no||
(4) || FM-DEFECT-RADIATOR-CAP
        INSTANCE : FAILURE-MODE
        DESCRIPTION : "Leak from defect radiator cap"
        HAS-TESTS : T-RADIATOR-PRESSURE-CAP-TEST
              (confirming-results BEYOND-LIMITS)
       has-repairs : rp-replace-radiator-cap
       related-parts : cap|
(5) || T-VISUAL-INSPECT-COOLANT-LEVEL
        INSTANCE : "Visually inspect coolant level"
        PROMPT : "Please visually inspect coolant level."
        POSSIBLE-RESULTS : Low normal||
```

(6) ¦¦ T-RADIATOR-PRESSURE-CAP-TEST
INSTANCE : test
DESCRIPTION : "Perform radiator pressure cap test"
HAS-TEST-TOOLS : Rotunda-21-0012-radiator-pressure-tester
PROMPT : "Please perform radiator pressure cap test as set out in «RADIATOR PRESSURE CAP TEST» p. 14
Are test readings within limits or beyond limits ? "
possible-results : WITHIN-LIMITS BEYOND-LIMITS¦¦

(7) ¦¦ RP-REPLACE-RADIATOR-CAP
INSTANCE : repair
DESCRIPTION : "Replace the radiator cap§"
HAS-PROCEDURE : "Please replace the radiator cap."¦¦

If engine overheat, customer concern failure mode, is confirmed, two questions of HAS-QUESTIONS slot are asked and attached meta rules are examined.

As an example, the following meta rule, written is OPS-5, is invoked when the engine overheat problem only occurs under severe driving conditions for non-front-wheel car, and rearranges the order of investigation of the causes for engine overheat by modification of DUE-TO list.

```
(p overheat ! not-air-cond-run ! severe-drive ! not-fr-wheel
  (questions ^instance questions
             ^in-failure-mode FM-ENGINE-OVERHEAT
             ^q-has-air-conditioner <has-air-conditioner>
             ^q-engine-overheat-when-air-conditioner-running
                    <<NO UNKNOWN>>
             ^q-engine-overheat-under-severe-driving-condition YES
             ^q-front-wheel-drive NO)
 ...>
  (new-values 'fm-engine-overheat
              'due-to
              '(fm-excessive-loads
                fm-low-coolant-level
                fm-lack-of-coolant-flow
                fm-air-flow-obstruction
                fm-contaminated-coolant
                fm-fan-problem
                fm-low-oil-level
                fm-improperly-installed-air-conditioner
                fm-other-engine-related-problems))
  (if (eq $<has-air-conditioner> 'NO)
     (delete-value 'fm-engine-overheat
                   'due-to
                   'fm-improperly-installed-air-conditioner)))
```

8. Summary

In this paper new approach to machine diagnosis has been described taking advantage of knowledge representation and inference techniques in expert systems. The focus is how to organize and represent the expert knowledge and mimic the human expert reasoning process.

Tests on the prototype DESACS system have shown that the system was able to correctly and efficiently track down the component causes of a failure in the automobile cooling system. What has been introduced here can be applicable to not only machine diagnosis domains but also medical, management and other diagnostic fields.

References

1. Bellows, J. C., "An Artificial Intelligence Chemistry Diagnostic System," *In Proceedings of the 45th International Water Conference*, Pittsburgh, PA, (1984).
2. Buchanan, B. G., and Shortliffe, E. H. (eds), *Rule-Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA : Addison-Wesley Pub. Co., (1984).
3. Clark, P. E., McShane. I.E., Wakeley, K., "The Diagnosis and Solution of Electrical Machine Problem : Design Influence," *IEEE Transactions on Industry Applications*, 23, (Jan/Feb 1987), pp. 28-36.
4. Davis, R., "Diagnostic Reasoning Based on Structure and Behavior," *Artificial Intelligence*h, 1984), pp. 347-410.
5. Folley, J. D. Jr, and Hritz, R. J., "Embedded AI Expert System Troubleshoots Automated Assembly," *Industrial Engineering*, 19, (April 1987), pp. 32-36.
6. Fox, M. S., Kleinosky, P., and Lowenfeld, S., "Techniques for Sensor-Based Diagnosis," *In Proceedings of he 8th International Joint Conference on Artificial Intelligence*, Menlo Park, CA, (1983), pp. 158-163.
7. Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. (eds), *Building Expert Systems*, Reading, MA : Addision-Wesley Pub. Co., (1983).
8. Larner, D. L., "A Recursive Expert Troubleshooting System Utilizing General and Specific Knowledge," *In Proceedings of the 2nd Conference on Artificial Intelligence Application*, (1985), pp. 34-41.
9. Osborne, R. L., "Online, Artificial Intelligence-Based Turbine Generator Diagnostics," *AI Magazine*, 7(4), (Fall 1986), pp. 97.
10. Pepper, j., "An Expert Systems for Automotive Diagnosis," *Ford SBDS*, (1986).
11. Pepper, J., and Mullins, D. R., "Artificial Intelligence applied to Audio Systems Diagnosis," *In Proceedings of the International Congress on Transportational Electronics*, Dearborn, MI, (Oct. 1986).
12. Reiter, R., "A Theory of Diagnosis from First Principle," *Artificial Intelligence*h, 32, (1987), pp. 57-95.
13. Silver, B., *Meta-Level Inference*, North-Holland, (1986).
14. Tanner, M. C., and Bylander, T., "Application of the CSRL Language to the Design of Expert Diagnosis System : The Auto Mech Experience," *In Artificial Intelligence in Maintenance*, (1985).
15. Waterman, D. A., *A Guide to Expert Systems*, Reading, MA : Addison-Wesley Pub. co., (1986).
16. *Diagnosis of Cooling Systems*, Certified Training Program CTP-1977-3, Ford Parts and Service Division, Training and Publication Department, Ford, (1977).
17. *Knowlwdge Craft, Ver 3. 1, Manuals*, Carnegie Group Inc., Pittsburgh, PA., (1986).