EXPLANATION-BASED LEARNING WITH DIAGNOSTIC MODELS

John W. Sheppard

ARINC Research Corporation, 2551 Riva Road, Annapolis, MD 21401
Phone: (410) 266-2099, Email: sheppard@cs.jhu.edu

ABSTRACT

Research in machine learning has provided a new avenue for developing models and knowledge bases for diagnosis. Learning approaches evolve models on actual examples of failure and diagnosis. Using these examples (and possibly an initial model of the system), diagnostic systems learn relationships between the tested system tested and the faults causing anomalous behavior. We discuss an approach to identifying and correcting errors in diagnostic models using explanation based learning. The approach uses a model of the system to be diagnosed that may be missing information about the relationships between tests and possible diagnoses. In particular, we will use what has become known as a structural model or information flow model to guide diagnosis. When misdiagnosis occurs, we will use the model to determine how to search for the actual fault through additional testing. When we finally identify the fault, we will construct an explanation from the original misdiagnosis and modify the model to compensate for the incorrect behavior of the system.

INTRODUCTION

Intelligent, computer assisted diagnosis has been a popular area of research for many years. Medical doctors have been interested in how to use the computer to consult in medical diagnosis. Maintenance technicians and testability engineers in airline maintenance shops and military maintenance depots have been searching for better ways for finding faults in the systems they maintain. Controllers in nuclear power plants have been looking for ways to use the computer to monitor and predict when failure may occur so that they can anticipate and correct the problem before it occurs. Artificial intelligence researchers have found that the general category of diagnosis offers fertile ground for exploring the issues of computer-based reasoning.

In response to these needs, several approaches to computer-based diagnosis have arisen. Shortliffe[1] demonstrated that rule-based systems could be applied successfully to complex medical problems with his MYCIN system. Davis[2] described a model based on the structure and behavior of a system for diagnosing digital circuits. More recently, Cantone, et. al[3], Simpson and Sheppard[4], and Pattipati and Alexandridis[5] have developed generic architectures for fault diagnosis using dependency-based structural models.

One area with tremendous research potential in computer-based diagnosis is in the area of developing the knowledge bases for the diagnostic systems. In the past, knowledge bases were constructed through a cycle of interviewing experts, coding rules, generating prototypes, and testing the prototypes against the experts. The process continues until the resulting diagnostic system performs according to some set of user requirements. Unfortunately, this approach is generally ad hoc and incomplete.

Other approaches include developing complex models of the system to be tested. Modelers study design documents and existing technical manuals to learn the "physics" of the system to be diagnosed. The resulting models then become the knowledge base for diagnosis. Though more rigorous in their development, models constructed in this way are highly prone to error due to the inherent complexity of the systems modeled. Also, with advances in technology, complexity is not going to decrease. Therefore, a need exists for controlling the complexity of the problem while still producing robust and efficient models for diagnosis.

Research in machine learning has provided a new avenue for developing models and knowledge bases for diagnosis. Resulting approaches evolve models based on actual examples of failure and diagnosis. Using these examples (and possible an initial model

CH3148-4/92/0000-0159 $1.00©1992 IEEE

of the system), diagnostic systems learn the relationships between the system being tested and the faults causing anomalous behavior.

The problem with this approach is that the evolution of models can span a long period of time. An alternative approach would be to begin with a relatively complete model produced by traditional methods and include a learning system that would enable the knowledge base to adapt when errors are identified. As a result, modelers can construct diagnostic systems relatively quickly. Complexity is reduced since the model can be incomplete. Then, through the process of actual diagnosis and repair, problems are identified and corrected through the interactions of a maintenance technician and the diagnostic system.

In this paper, we will discuss an approach to identifying and correcting errors in diagnostic models using explanation-based learning. The approach uses a structural model of a system to be diagnosed that may be missing information about the relationships between tests and possible diagnoses. When a misdiagnosis occurs, the approach continues to search the model for the fault through testing. When the fault is finally identified, an explanation for the original misdiagnosis is generated, and the model is modified to compensate for the incorrect behavior of the system.

MODEL-BASED FAULT DIAGNOSIS

Fault diagnosis has evolved from analyzing simple combinational circuits to analyzing complex systems consisting of combinations of electrical, mechanical, and even hydraulic systems. Diagnostic systems are being developed that monitor the performance of avionic systems in aircraft and reconfigure the avionics suite when a fault occurs. The task of fault diagnosis is to detect when a fault occurs and then, by performing a sequence of tests, isolates the failed element in the system.

In model-based diagnosis, system specifications and design data can be used to define a mathematical model. This model includes information about the structure or the behavior of the system. In more advanced models, both structural information and behavior information are provided. This model then becomes the knowledge base in the diagnostic system. As a result, analysts and engineers can build diagnostic models without a maintenance expert, and new and complex systems can be diagnosed efficiently and effectively. A discussion of the different types of models can be found in reference 6.

LEARNING IN FAULT DIAGNOSIS

A problem common to all computer-aided diagnostic systems is that knowledge bases (whether rule-bases or models) are difficult to develop. As a result, errors are common. This leads to inefficient and even incorrect diagnosis. Further, as the complexity of systems increase, the likelihood of erroneous models increases. The questions that naturally follow from this problem include:

1. How does one develop models that minimize the chance of error?

2. In the event errors occur, how does one identify and correct the errors?

Work in the area of machine learning provides potential solutions to both of these problems. Through a process of simulation or fault insertion, examples can be generated in which the failed behavior is captured and its relationship to the available tests is determined. Also, discrepancies in repair recommendations and actions taken to repair the system can be used to identify the errors in the model. In addition to determining the structure of the system, learning can be applied to improving diagnostic performance.

Learning Optimization Parameters

One of the advantages of model-based diagnosis is that the diagnostic process can be optimized. Unfortunately, the process of determining the most efficient diagnostic strategy for an arbitrary system is NP-Complete[7]. Nevertheless, some steps can be taken to improve diagnosis. In particular, Pattipati and Alexandridis[5] and Simpson and Sheppard[4] have described algorithms using Shannon's information theory to optimize binary decision trees. These algorithms select tests that maximize information gained per unit cost where cost may be given in terms of combinations of time, failure probability, etc.

When applying cost criteria to optimize diagnosis, diagnostic performance can be improved by updating cost estimates. Following diagnosis the reported costs can be used to modify the original

weights thereby modifying the optimization procedure to reflect actual diagnostic history. A detailed discussion of our approach to learning optimization parameters is given in reference 6.

Connectionist Learning

Another approach to learning that is becoming popular in fault diagnosis involves connectionism (i.e., neural networks). Diagnostic neural networks map test results to diagnoses through a process of training. When training a diagnostic neural network, the network processes several example test-fault combinations and learns by minimizing the error in its output through a hill-climbing algorithm.[8] Neural networks have been developed that map test results (inputs) to diagnoses (outputs).[9] In addition, neural networks have been trained to interpret signals generated from running a test to determine if a test has passed or failed.[10] Neural networks have also been developed to interpret the results of the inference process (under uncertainty) to determine if additional testing is necessary.[11]

Similarity-Based Learning

The learning method applied for neural networks frequently falls in the category of similarity-based learning (SBL). DeJong describes SBL as "discovering a combination of features that best classifies the regularities in a set of examples. The resulting generalization over the examples is the new concept[12]." In general, SBL is characterized by presenting several examples and by not requiring much domain knowledge. The training examples contain most of the required knowledge.

Explanation-Based Learning

In the event domain knowledge is available, this knowledge can be used to reduce the number of training instances in learning. Explanation-based learning (EBL) is characterized by using a detailed domain theory and a detailed functional specification of the concepts to be learned. Using the domain knowledge, single training examples can be used to learn concepts[12]. In the domain of fault diagnosis, and example of a misdiagnosis together with a model of the physics for the technology employed in the system can be used to derive an explanation of an appropriate diagnosis. From this explanation, the diagnostic model can be modified to include the knowledge of the correct diagnosis.

The remainder of this paper will be devoted to a discussion of an approach to using EBL in fault diagnosis. The approach uses a theory of fault diagnosis based on a structural model in which the physics of diagnosis is specified in the test design. This permits the problem to be reduced to identifying appropriate dependency relationships between tests and components.

LEARNING STRUCTURAL MODELS USING EXPLANATION-BASED LEARNING

One approach to incorporating explanation-based learning in diagnosis takes advantage of the form of the structural model. The central idea behind this approach involves following a misdiagnosis with additional testing until a correct diagnosis is made. Once the correct diagnosis has been made, the knowledge obtained from testing can be used to modify the structure of the model so that the correct diagnosis is consistent with the testing. Ultimately, this should lead to a correct model.

Assumptions for the Model

As indicated above, this study applied EBL to structural diagnostic models. Inherent in these models and in the approach are several assumptions. For the purposes of the following discussion, we define the following notation and concepts. In these discussions, we will associate a conclusion to be drawn with each component in the model. We will use the terms *conclusion* and *component* interchangeably. Let

- M = A correct input model consisting of first order dependencies.
- M^* = The transitive and logical closure of M.
- M_ϵ = An input model containing the error(s).
- M_ϵ^* = The transitive and logical closure of M_ϵ.
- c_{fail} = The conclusion associated with a failed component in the system.
- c_{isol} = The conclusion associated with the isolated component in the model.
- c_i = The conclusion associated with the i^{th} component in the model.
- t_j = The j^{th} test in the model.

We also define the following.

Def.1: A test t_i *depends on a component* c_j iff when c_j fails, t_i will be *bad*.

Def.2: A test t_i *depends on a test* t_j iff when t_j is *bad*, t_i is also *bad* and when t_i is *good*, t_j is also *good*.

Def.3: A conclusion c_i is a *nondetection* iff no tests depend on c_i.

Def.4: An *ambiguity group* is a set of components in either M^* or M_e^* such that the set of tests that depend on each component in the ambiguity group is identical.

Given these definitions, we wish to develop a strategy to apply to fault diagnosis in which an erroneous or incomplete model can be modified to correct or complete the model. In other words, as a result of performing a sequence of tests in which c_{incol} is incorrectly isolated, we wish to perform additional tests so as to isolate c_{fail} thus enabling M_e^* to be transformed into M^*. As a first step in addressing this problem, we will consider the problem under the following assumptions.

First, as is typical in fault diagnosis, we assume a single failure exists in the system. As described earlier, this limits the search space; although, certain extensions to the model and to the inference rules applied to the model permit this assumption to be relaxed. Second, we assume that the tests specified for the model provide complete and accurate information about the system. This means that if we say that a test depends on a component, then if the component fails, the test will detect the failure, and if the test passes, then the component has not failed. Finally, we assume that only tests have dependencies, and that they can only depend on components or other tests. To say that a component depends on another component or on a test (from the structural perspective) makes no sense. This is because tests are simply information carriers and have no impact on the behavior of the system it is testing.

Inference Rules

The diagnostic system we will use to implement the described learning approach incorporates several inference rules specifically tailored to the diagnosis problem. In particular, the following rules are used.

Rule 1: If t_i is declared to be *untestable*, then make t_i unavailable for evaluation.

Rule 2: If t_i is declared to be *good*, then declare every test upon which t_i depends to be *good*.

Rule 3: If t_i is declared to be *good*, then declare every conclusion upon which t_i depends to be *false*.

Rule 4: If t_i is declared to be *bad*, then declare every test that depends on t_i to be *bad*.

Rule 5: If t_i is declared to be *bad* and there exists a test t_j that neither depends on t_i nor is depended on by t_i, and the elimination of t_j does not create additional ambiguity, then declare t_j to be *not needed*.

Rule 6: If t_i is declared to be *bad*, then declare every conclusion that t_i does not depend on and that has no test declared *not needed* that depends on it to be *false*.

Rule 7: If t_i is declared to be *bad*, then declare every conclusion that t_i does not depend on and that has at least one test declared *not needed* that depends on it to be *not relevant*.

Rule 8: If t_i depends on *all* of the unknown conclusion, then declare t_i to be *bad*.

Rule 9: If t_i depends on only *false* conclusions, then declare t_i to be *good*.

Identifying Missing Structural Links

To determine which dependency links have been omitted from the model, we need to develop a well-defined approach to identifying the correct fault following an inappropriate fault isolation. To do this, we proved the following claims (proofs omitted do to space limitations).

Claim 1: If a failure, c_{fail} is detected, then there exists a test t_i whose outcome is *bad* that depends on c_{fail} in M^* that was evaluated.

This claim should be self evident. Simply, in order for a failure to be detected, there must exist some test whose outcome is bad when the failure occurs.

Let us partition the model M_e^* following fault isolation according to inferences or measurements made on tests and corresponding values associated with conclusions.

$$\Pi_1 = \{ t_i \mid \text{val}(t_i) = \textit{good} \} \cup \{ c_i \mid \text{val}(c_i) = \textit{good} \text{ [Rule 3]} \}$$

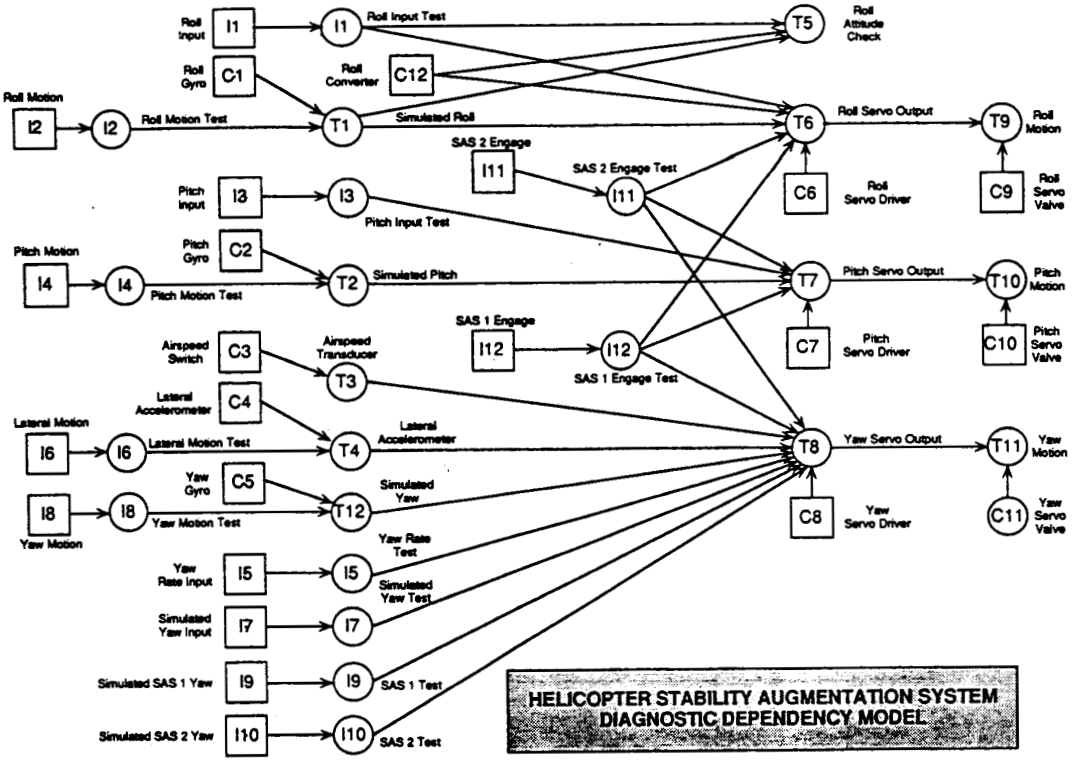

Figure 1. Helicopter Stability Augmentation System Diagnostic Dependency Model

$$\Pi_2 = \{ t_i \mid \text{val}(t_i) = \textit{bad} \} \cup \{ c_i \mid (\text{val}(c_i) = \textit{unknown}) \vee (\text{val}(c_i) = \textit{good} \text{ [Rule 6]}) \}$$

$$\Pi_3 = \{ t_i \mid \text{val}(t_i) = \textit{not needed} \} \cup \{ c_i \mid \text{val}(c_i) = \textit{not relevant} \}$$

Claim 2: Given the model M_e^*, if there exists a missing dependency link, then some test t_i that was declared *bad* must depend on some conclusion $c_{fail} \neq c_{isol}$ in M^* that was declared *not relevant*.

Recall that M_e^* represents the model that is missing the dependency link. Thus if $c_{fail} \neq c_{isol}$ we must have isolated the wrong failure. This means c_{isol} was eliminated from consideration. A component can be eliminated from consideration by being declared *false* or *not relevant*. A *false* declaration only occurs by either *Rule 3* or *Rule 6*. This claim shows that neither of these are possible.

Obs. 1: When all tests have some value assignments (i.e. *good*, *bad*, or *not needed*), then the drawn conclusion, c_{isol}, is the conclusion (or ambiguity group) that is still unknown (i.e. does not have value assignments of *false* or *not relevant*).

This observation follows directly from the fact that the set of unknown conclusions is, by definition, the set of candidate failures. Thus, when all other information is known, the remaining candidate set must be the isolation.

Obs. 2: Let S be the set of conclusions c_i that have all *bad* tests depending on them. c_{isol} is in S.

Since the model specifies which tests are expected to fail given a fault in the system, then the drawn conclusion must have all of the bad tests depending on it. If any test is bad and does not depend on the drawn conclusion, some other failure must have been detected in the system.

Obs. 3: Let F_i be the set of tests that depend on conclusion c_i. Then $|F_{isol}| = max\{|F_i|, \forall c_i \in S\}$.

Obviously, since *all* of the bad tests depend on the drawn conclusion, any other conclusion in which all of its dependent are bad tests must have fewer dependents (unless that conclusion is ambiguous with the drawn conclusion). Using these observations, we can make the following claim.

Claim 3: The head of the missing link enters a test that depends on c_{isol} yet does not depend on any other conclusion in Π_2.

With this claim, we have identified the best point in which to attach one end of the missing dependency. The proof shows that any other appropriate location will still leave the model with missing links, even if these locations are consistent. In the worst case, any other location may create an inappropriate link.

Claim 4: Given an incorrect isolation, c_{isol}, fault isolating in partition Π_3 will result in isolating c_{fail}.

This claim allows us to identify which partition contains the actual fault. Clearly, c_{fail} cannot be in Π_1 since a test that *does* depend on c_{fail} would have eliminated c_{fail} from consideration. This is a contradiction. Further, c_{fail} cannot be in Π_2 since this would be c_{isol}. Now we only need to find where in Π_3 the tail of the link needs to be placed.

Claim 5: The tail of the missing link is on the path from c_{fail}.

This claim is obvious but is stated (and has been proven) for completeness.

Obs. 4: Let C_i be the set of conclusions that t_i depends on and let C_j be the set of conclusions that t_j depends on. If $C_i \subseteq C_j$ then t_j depends on t_i.

This observation allows us to prove the correctness of an algorithm called *logical closure*.[13] Effectively, this algorithm allows us to limit the specification of new dependencies to be between tests and conclusions. Test-to-test dependencies are inferred from test-to-conclusion dependencies.

Claim 6: An iterative application of isolating proper faults given a missing link will reduce Π_3 and approach identifying the missing link.

This claim is one which declares that the process will eventually terminate with the correct model, M'. As such, it is an important claim that declares the completeness of the algorithm.

Claim 7: When a link is added to the model, it does not need to be removed, even if it is not the required link.

To prove the soundness of the algorithm, we had to show that no inappropriate links would be specified. This claim serves that function.

An Example

In order to illustrate the procedure described in the previous section, we developed a diagnostic model of the stability augmentation system for a military helicopter. See Figure 1 for a pictorial representation of this model. Circles in the figure correspond to tests in the model, and squares correspond to components. The arrows indicate the flow of failure information through the system. For example, if the Roll Converter fails, then the Roll Attitude Check, the Roll Servo Output, and the Roll Motion tests will all be bad.

As an example of how one might find a missing link, suppose the model should include a dependency of the Airspeed Transducer test on the Roll Converter. The following sequence of tests will inappropriately isolate the Airspeed Switch as the failure.

Test: Yaw Servo Output declared bad.
Test: Lateral Accelerometer declared good.
Test: Simulated Yaw declared good.
Test: Airspeed Transducer declared bad.
Component Airspeed Switch isolated.

Once we determine that we isolated the wrong component, we perform additional tests to isolate the correct fault–the Roll Converter.

Test: Roll Motion Output declared bad.
Test: Roll Attitude Check declared bad.
Test: Simulated Roll declared good.
Test: Roll Input Test declared good.
Component Roll Converter isolated.

As a result of the testing performed to isolate the Roll Converter, we can determine that the Yaw Servo Output test, the Airspeed Transducer test, the Roll Motion Output test, and the Roll Attitude Check test must depend on the Roll Converter. The dependence of the Airspeed Transducer test on

the Roll Converter is the missing link, and the dependence of the Yaw Servo Output can be determined from the transitivity of dependence. The other two dependencies already existed in the model. Thus, we correctly identified the missing link.

FUTURE WORK

This paper has described an approach to identifying missing dependency links in structural diagnostic models using an approach to explanation-based learning. Using the results of diagnostic testing, explanations for discrepancies in fault isolation and failure are generated in the form of needed dependencies. These needed dependencies are then used to determine which dependencies are missing. This section describes two additional applications areas based on technique.

Identifying Inappropriate Structural Links

A difficult problem to be considered is the one where errors in the model may include additional, inappropriate dependency links. We can partition the system in a way similar to the partitions for missing links, but we face the problem of what to do when the extra link is to a position downstream from the isolated failure. It appears that identifying the fault in this situation may result in a directed search down the dependency chain, considering each component in sequence.

Another interesting problem, given an algorithm for identifying extra links can be developed, is how to combine the two techniques. Initial research suggests that the first step should be to identify missing links. Since this procedure has been proven to terminate (under the assumptions described), if the correct component is not isolated, then we can recycle and use the "extra link" algorithm. This procedure would begin from the point the initial fault isolation was completed.

Creating Structural Models through Learning

Ultimately, we would like to develop a means for easily developing complete and robust models. If we can associate some test with each component in a system, then we can specify this as an initial structural model. This model has several missing dependency links and no extra links. Then through a process of fault insertion and fault isolation, we can identify additional dependency links until a complete, functional model is generated.

Alternatively, we could begin by claiming each test depends on all the components. Then we could apply the "extra link" algorithm to determine which links should be removed. Again, through a process of fault insertion and fault isolation, the extra links are gradually identified and removed until a complete and functional model results.

CONCLUSION

This paper has considered an approach to explanation-based learning in which erroneous or incomplete structural diagnostic models are corrected through the standard fault isolation process. When an inappropriate fault is isolated, additional tests are performed until the correct fault is isolated. Since tests are assumed to provide complete and reliable information (which has been reasonable for many "real" diagnostic problems), the test results indicate where dependencies are appropriate and where they are inappropriate. Whenever, such designation disagrees with the model, a change to the model is warranted.

Several issues related to diagnostic learning need to be considered. The first is model configuration control. If the model is non-stationary, how does one control the configuration of the model used by technicians in the field? In addition, how does one verify that the modified model is correct? (Obviously, if the approach has a proof of correctness such as the one described here, then this problem is straightforward to consider.) Finally, should the diagnostic system modify the model directly, or should the system offer recommendations to be considered for modification? Although not directly pertinent to the problem of learning diagnostic models, these issues must be considered for any system that is to move from the laboratory into practice.

REFERENCES

1. E. H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*, New York, Elsevier, 1976.

2. R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," *Artificial Intelligence*, Vol 24, 1984.

3. Richard R. Cantone, Frank J. Pipitone, W. Brent Lander, and Michael P. Marrone, "Model-Based Probabilistic Reasoning for Electronics Troubleshooting," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983.

4. William R. Simpson and John W. Sheppard, "System Complexity and Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 8, No. 3, September 1991.

5. Krishna R. Pattipati and Mark G. Alexandridis, "Application of Heuristic Search and Information Theory to Sequential Fault Diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 4, July/August 1990.

6. John W. Sheppard and William R. Simpson, "Elements of Machine Learning in a Field Diagnostic Maintenance Aid," *Proceedings of the ADPA Symposium on Artificial Intelligence Applications*, Williamsburg, Virginia, March 1992.

7. Laurent Hyafil and Ronald L. Rivest, "Constructing Optimal Binary Decision Trees is NP-Complete," *Information Processing Letters*, Vol. 5, No. 1, May 1976.

8. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, The MIT Press, Cambridge, Massachusetts, 1986.

9. K. A. Marko, J. James, J. Dosdall, and J. Murphy, "Automotive Control System Diagnostics Using Neural Nets for Rapid Pattern Classification of Large Data Sets," *Proceedings of IJCNN-89*, Washington, DC, June 1989.

10. W. T. Katz and M. B. Merickel, "Translation-Invariant Aorta Segmentation from Magnetic Resonance Images," *Proceedings of IJCNN-89*, Washington, DC, June 1989.

11. John W. Sheppard and William R. Simpson, "A Neural Network for Evaluating Diagnostic Evidence," *Proceedings of the National Aerospace & Electronics Conference*, Dayton, Ohio, May 1991.

12. Gerald DeJong, "An Introduction to Explanation-Based Learning," in *Exploring Artificial Intelligence*, Morgan Kaufmann Publishers, Palo Alto, California, 1988.

13. John W. Sheppard and William R. Simpson, "A Mathematical Model for Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 8, No. 4, December 1991, pp. 25-38.