Specialized Strategies: An Alternative to First Principles in Diagnostic Problem Solving*

Nancy E. Reed, Elizabeth R. Stuck and **James B. Moen**
Computer Science Department
University of Minnesota
Minneapolis, Minnesota 55455

Abstract

We introduce specialized strategies, an alternative level of reasoning, falling in generality between recognition-based reasoning and reasoning from first principles. These strategies are weak methods that are specific to a class of problems that occur in different domains. Specialized strategies are applicable not only to familiar problems in a domain, but also to problems that have not been anticipated. As a result they can provide both broad coverage currently given by "causal" reasoning and an efficiency close to that of "shallow" reasoning. The specialized strategies use inexact models of the components in the faulty system which contain only diagnostically relevant knowledge. Specialized strategies may be used in expert systems to increase efficiency, reduce brittleness, and decrease knowledge base construction effort compared to other common approaches. Examples are given from the domain of computer hardware diagnosis where two prototype expert systems were implemented.

1 Introduction.

Two approaches are commonly used in diagnostic expert systems: empirical associations or "shallow" reasoning, and reasoning from first principles or "causal" reasoning [Hart, 1982]. Diagnosis using shallow reasoning requires the complete specification of pattern $\rightarrow$ action knowledge to provide coverage. In addition, this knowledge must be acquired for every task. Diagnosis from first principles uses a complete description of the design of the system and the functionality of the components in it. The first principles approach provides complete diagnostic coverage of a task and this knowledge is readily transferred to new tasks.

A common limitation of both shallow reasoning and reasoning from first principles is the extensive initial knowledge specification necessary. The application of expert systems to the diagnosis of complex, changing, or short-lived systems is difficult as a result. We argue that expert systems can perform effective and efficient diagnosis with an inexact model of the system using specialized strategies.

Specialized strategies use a type of informal, qualitative, causal reasoning that is specialized to diagnose complex systems made up of many small, replaceable, connected components. Specialized strategies use inexact models of the components in the system. These models contain only the diagnostically relevant structural, functional, and fault knowledge. Complex systems may be diagnosed without a complete representation of the exact functioning of each of its components by focusing attention only on a small, localized part of the system at any one time. When inexact models are used, simulation of complete system performance at a global level is not possible. However, it is possible to perform local qualitative simulations and inferences of sufficient power to determine the location of faults.

Four specialized strategies we have found useful in computer hardware diagnosis are compare and conquer, heuristic path following, stateless analysis, and endpoint analysis. *Compare and conquer* compares one or more data values with reference values. This technique is useful when a component's function is so complex that its correct behavior is not easily deduced. *Heuristic path following* reduces the complexity of determining which component to examine next by using local information. This strategy helps to focus attention on the relevant portions of the system. *Stateless analysis* verifies the behavior of a component with internal state while ignoring some of the timing information associated with components of this type. This strategy is significant because it avoids the need to consider the global state of the system. *Endpoint analysis* may be used to locate a useful symptom of a fault when other information is not available. Attention is directed toward the components at the interface between the module being examined and the rest of the system.

These strategies are based on studies of experts in two different industrial troubleshooting environments. The methodology used for analyzing the expert behavior included directed interviews and verbal protocol analysis, similar to that in [Johnson *et al.*, 1987].

2 Related Work.

One approach to diagnosis uses first principles. Work in this area includes that of Genesereth [1984], Davis [1984], de Kleer and Williams [1987], and Reiter [1987]. Genesereth [1984] proposed the use of design descriptions to generate tests that could prove the correct or incorrect functioning of a component. This approach relies on the availability of complete design descriptions that are tuned to the diagnosis task. It does not take advantage of already existing tests. The major problem with this approach is that all possible fault types must be explicitly enumerated. Computational complexity is then reduced by eliminating fault types from consideration, which limits the types of

*This work was supported by IBM, the Control Data Corporation, and the University of Minnesota's Microelectronic and Information Sciences (MEIS) Center.

problems that can be solved.

Davis [1984] agreed that the above approach helped to determine whether or not a component was functioning correctly, but pointed out that an even more crucial task in diagnosis is finding the suspected faulty components to test. He introduced "pathways of causal interaction" as a key concept in localizing suspected faulty components. Complete information availability is assumed without cost. Since this approach also represents fault types explicitly, Davis was faced with a similar problem of computational complexity. His solution was to make simplifying assumptions to keep the problem tractable, then relax the assumptions as necessary. Davis suggested the use of incomplete models to help address the scaling issue and to generalize to other domains.

De Kleer and Williams [1987] used a model-based reasoning strategy with probabilistic information and a sequential probing strategy. This strategy is capable of diagnosing multiple faults, but is limited to faulty components (bridge faults are excluded, for example). Individual fault rates of components are used. A best-first search strategy guides the acquisition of relevant information.

Reiter [1987] developed a domain-independent mathematical theory of diagnosis from first principles. In order to construct such a diagnostic model, it is necessary to specify a finite set of disorders (faults), manifestations (symptoms), and causal connections (from symptoms to faults). An algorithm for diagnosis is given for systems specified in this manner.

First principles approaches are currently computationally expensive and limited to diagnosing a few types of faults. They cannot easily diagnose systems with many components, complex components, or components with state. In addition, these approaches rarely take advantage of available knowledge, such as the fault rates of components, available tests and tools, ease of testability, and the interaction between repair and test. However, the above work demonstrates the usefulness of design and/or functional knowledge in diagnosis.

3 Problem Class Characteristics.

The specialized strategies discussed in this paper are applicable to a class of problems. These problems include the diagnosis of systems with the following characteristics. The system has a large, complex organization, composed of many connected components, each of which is replaceable and/or repairable. There exists a flow of data between components that may be measured at many different points [Figure 1]. Information available at the start of the diagnosis is usually insufficient to determine the fault(s), necessitating the acquisition of more information. The number of fault types is small, but each may occur in many different locations, so that a solution consists of both a fault type and its location. The symptoms of a fault are dependent upon the particular function and type of the faulty component, as well as the location and type of fault. The lifetime of the system may be short, limiting the time available to build a diagnostic expert system.

Many systems have these characteristics, including computer hardware systems and other complex machines. Large software programs have similar characteristics. The components in this domain are modules of code that interact through invocation and shared data structures.

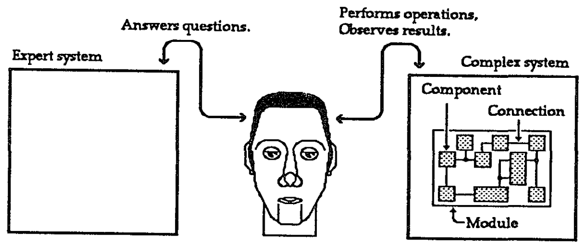

Figure 1: An expert system in the diagnosis process.

3.1 Diagnosis Process.

Since insufficient information is present at the start of the diagnostic process, this is a sequential diagnosis task [Gorry and Barnett, 1968]. Other relevant information must be obtained during the process of diagnosis to assist in the solution of the fault. Data is obtained by performing three types of *operations*: information operations, data operations, and repair operations. *Information operations* obtain information that describes the correct functioning of a system or component. *Data operations* measure data flowing between components. *Repair operations* repair faults. Repair is not separate from diagnosis. A repair operation is always followed by a test to verify the repair and to check for other possible faults. Specific tests are used to set up *contexts* within which to obtain data.

Each operation has a cost based on the time and resources necessary to complete it. Many operations may potentially be performed at each step, making exhaustive data collection prohibitively expensive. Knowledge may be used to determine the best operation to perform next. Time constraints based on the relative value of the system and its components require the diagnosis to be efficient. The expert system in Figure 1 assists the user during diagnosis by suggesting useful operations to perform. The expert system could also be applied directly to the complex system using special interfaces.

3.2 Example domain.

Examples given are from the domain of computer hardware diagnosis, specifically *board level diagnosis*. A computer board (module in the figure) consists of various electrical components and connections (links) between them. The goal is to isolate the smallest repairable/replaceable component while minimizing the resources used and the time spent. The most common types of faults are broken connections, shorts between connections or components, and malfunctioning components. We assume single, nonintermittent faults.

Three types of resources are available for use in board level diagnosis. *Information sources* describe the physical and functional properties of the board and/or system. They are used by information operations. *Tools*, such as a soldering iron, are used in repair operations. Other tools, such as an oscilloscope, are used in data operations. An example data operation is using an oscilloscope to probe a particular point on a board to obtain the resulting waveform. Finally, special *tests* may be available that exercise specific functions or components of the system. Some contexts are defined by applying a set of signals to the input tabs of the board (using a test). Other contexts include positioning the board to perform a continuity test or positioning at a microscope for visual inspection.

4 Specialized Strategies.

The following sections describe the strategies and inexact models used by experts in computer hardware diagnosis. We have implemented some of these strategies and models in two prototype expert systems.

Specialized strategies are directed by high-level control strategies. One example is *difference pursuit*, a general high-level control strategy which has four steps. First, a context that includes a failure is established. This context of failure may be a particular set of data sent through the system. Next, an observable difference associated with the failure is identified. Then the point where the difference first appears is identified. Finally, local testing is done to determine the component responsible for the fault. The first two steps in difference pursuit are exploratory and produce a symptom of the fault (*exploration* phase). Global information, from functional or test documentation, is very useful in these steps. The last two steps localize a specific fault (*localization* phase). This may be accomplished using almost exclusively knowledge about local components. Difference pursuit is similar to Davis' [1984] "violated expectations", but also has steps that create the context of failure before a difference can be observed. Creating this failure context may be a difficult task to which test generation, as discussed by Genesereth [1984] may be applied.

Control strategies use the four specialized strategies for exploration and/or localization. These strategies direct data acquisition by suggesting useful operations to perform. Other control strategies may be useful for specific problems. The diagnosis of commonly occurring faults may be compiled into a sequence of operations in the form of a fault isolation tree.

4.1 Compare and Conquer.

Compare and conquer is used to determine if data measured from the system is correct or incorrect. The data is compared to a reference value which may be obtained from a correctly functioning component in the same context, or a specification of the component. Compare and conquer can be used for finding an initial symptom (exploration) or localizing the fault.

Compare and conquer circumvents the need for a high-level understanding of the system being diagnosed. In addition, this strategy lessens the need for a detailed low-level understanding of the system's components. For complex components or complex contexts, it is much easier to obtain the correct value from a working component than it is to simulate the component in the context. Comparing functionally equivalent components can also be used to eliminate error due to variability in measuring instruments. This strategy is used to determine what data values are correct. The other three strategies are concerned with where to look for data that might be incorrect.

4.2 Heuristic Path Following.

Diagnosis of complex systems is possible only if attention can be focused on portions of the system directly related to a given fault. For large systems, it is not possible to predetermine a complete set of *triggering rules* [Thompson *et al.*, 1983] that focus attention based exclusively on initial symptoms. The heuristic path following strategy uses a general understanding of the context and a symptom of the fault to track the symptom backwards until the failing component is located. Heuristic path following is primarily a localization strategy, since it eliminates irrelevant portions of the system from investigation.

The main problem that path following solves is determining which component to examine next. This decision is based on the initial problem symptoms, the path of the incorrect data, the context, and the behavior of the component being examined. This information determines which components and paths are relevant. As the path following strategy tracks the data from component to component, only those components that are relevant are examined. A path is relevant if it has influenced or determined the incorrect data in some way. For example, in board level diagnosis, if the signal being tracked comes from a multiplexer, the `select` signals of the multiplexer become relevant, since they determine which input of the multiplexer should be tracked.

There are two variations of the path-following strategy. The *single-stepping* method tracks data through the components, one by one. The *subdivision* method works much like binary search. When data measured at one component has a correct value and data at a second component has an incorrect value, data values are obtained at intermediate components, progressively narrowing the distance between the correct and incorrect values.

4.3 Stateless analysis.

The operation of some components may depend on their internal states. Stateless analysis is used for testing suspected failing components with state. It is primarily a localization strategy. Detailed reasoning about the behavior of such components can involve difficult temporal reasoning. Diagnosis can often be performed without explicit reasoning involving time, even for components containing internal state information. This is possible using a qualitative description of temporally varying data and comparing these descriptions without any reference to an absolute time standard.

Stateless analysis is successful because not all timing information is ignored. It uses qualitative timing information to determine at what moment or state the component should be examined. This moment is specified by its relation to the timing of the test being executed. The stateless analysis strategy determines the appropriate moment by using compare and conquer to find the time at which the data differs from what it should be. This instant is then used as the time at which all data paths relevant to this particular component are to be examined and compared. This approach works because it selects the most informative state of the component – the state in which the error occurs – at which to investigate its behavior.

4.4 Endpoint Analysis.

Endpoint analysis is an exploration strategy and is used to obtain an observable symptom that may be localized. Attention is focused on the input and output paths of the module being diagnosed. Compare and conquer is often used to determine if the data obtained is correct or incorrect. Endpoint analysis does not depend on any detailed knowledge of the system or the tests available. It may be performed more efficiently with information about which input or output paths are relevant, however.

A variant of this strategy is called *Easter egging.* This strategy uses a near random search for invalid or incorrect data. Its effectiveness depends on knowing what kind of data values are invalid or using compare and conquer to determine if a data value is incorrect. Endpoint analysis leads to a starting point for more focused search.

5 Inexact Models.

The knowledge contained in the inexact models is used together with the specialized strategies to perform diagnosis. The collected knowledge about each type of component is called a model of that type of component. The inexact model of the system contains many of these models, each describing properties of one class or type of component. The key characteristic of each model is that it contains only diagnostically relevant information. Information used only for design or simulation is not necessary. For example, the functioning of a complex component is not diagnostically relevant if a working component of the same type is always available for comparison. These models are inexact in that they are imprecise and incomplete views of the system being diagnosed.

The knowledge incorporated in these models is of three types: structural, fault, and functional knowledge. *Structural knowledge* describes how components should be connected to each other.

Fault knowledge describes what types of faults may occur, how to repair each fault type, and how frequently different fault types occur. Fault knowledge also describes what can be observed and what can be tested in a particular system, and when to use specific tests and tools. This category also includes knowledge describing *valid* and *correct* data values. A data value must be within one of the specified ranges to be valid. This range depends on the type of component. Any value inside the accepted range is valid. This makes the detection of a valid value a qualitative, rather than quantitative, decision. A valid data value may be correct or incorrect depending on the context.

Functional knowledge describes how components should behave. This category includes some limited causal knowledge. Constraints specify a component's correct behavior. The constraints may be defined in terms of what a component's input data should be, given its output data, and vice versa. Certain components or paths may also be constrained to propagate only specific data values. If the constraints on a component's behavior are not met, the component may be malfunctioning.

In computer hardware diagnosis, an inexact model of a connection (link) makes use of the knowledge that links connect components by transmitting signals from one end to the other. Therefore the same signal should appear on both ends of the link (or all ends for an n-way tee). Also, a continuity test of the link should yield the value `continuous`. Fault knowledge associated with a link describes two common faults, opens and shorts. An *open* is when the ends aren't (completely) connected. A *short* is when an adjacent link or component is connected when it should not be. The repair for an open is to add a wire. The repair for a short is to remove the solder or trace.

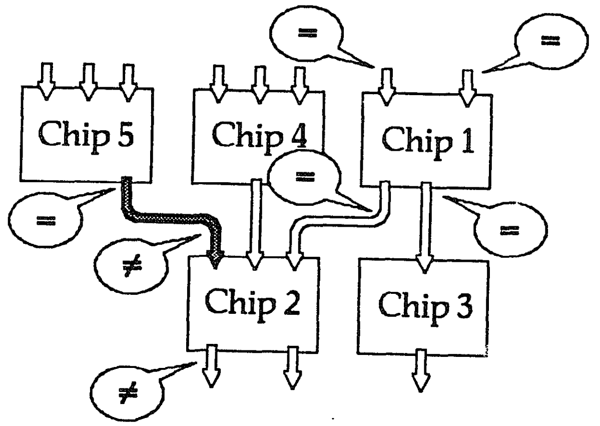

Figure 2: A portion of the board examined in the example.

6 Example.

Combinations of the strategies discussed above are used to diagnose a fault. A test that produces an error provides the context and may also provide focus information, highlighting components or modules for examination. In the absence of such information from the test, endpoint analysis can be used to find an initial symptom of the fault. Data is obtained until an incorrect value is observed. Compare and conquer is useful in determining if a data value is correct or incorrect. The inexact model of the system is used both to determine the expected values of the component's inputs and outputs, and to judge whether those values are reasonable. Once a difference has been found, path following is used to determine which path to follow and which component to examine next. Any components that have influenced the data either directly or indirectly are relevant to the path following strategy. The stateless analysis strategy may be used to examine components with state. Diagnosis proceeds, examining components and following paths until the point where the symptom appears is located. Then local testing is performed to distinguish possible fault candidates.

The following example describes the detection of an open in a link using specialized strategies [See Figure 2]. The symptoms of a diagnostic test initially focus attention on one chip, Chip1. Since Chip1 has a complex behavior, its function is verified by comparing its inputs and outputs with those of a similar correctly functioning chip. This is done by measuring the signals on Chip1's pins, and measuring the signals on a good chip in the same testing context. Since both chips are found to have the same signals on all pins (noted with "=" in the figure), no difference is detected and Chip1 is assumed to be working correctly.

Because of the particular test being executed, it is known that Chip2 may also be relevant, but not Chip3, so Chip2 is examined next. Functional knowledge of Chip2 indicates that an output signal should be a clock signal. The signal on this output pin is obtained, but is not in the range for the correct clock signal (noted with "$\neq$" in the figure). Since a difference in the clock output of Chip2 has been detected, path following determines that the corresponding clock input of Chip2 should be examined next rather than the other inputs from Chip4 or Chip1. The signal obtained on this clock input is also incorrect. Thus, Chip5, which is the immediate source of this clock signal to Chip2, is known to be relevant. The corresponding output on Chip5 is examined, and a correct clock signal is obtained. Conflicting signals on the ends of the link between Chip5 and Chip2 are detected. As a result, an open (fault) is suspected in the link. To verify this hypothesis, a continuity test of the link is made. The continuity test fails, and the fault in the link is verified. Adding a wire repairs the fault.

7 Implementation and Discussion.

Specialized strategies and inexact models have been implemented in two prototype expert systems. These prototypes diagnose faults in boards from two different computer systems. Despite large differences between the two systems, the strategies were successfully used to diagnose several faults, directly demonstrating generality across systems. The four specialized strategies discussed above may not be an exhaustive list of those useful in computer hardware diagnosis. An investigation of other strategies is a direction for future research.

The four specialized strategies are not limited to the domain of computer hardware diagnosis, but may also prove to be useful in diagnosing other systems with the similar characteristics of a complex organization, made up of many small connected components. Different domains may provide additional strategies and/or alternative ones. Specialized strategies may also be useful for reasoning at multiple levels of abstraction, although this has not been investigated.

Specialized strategies have several advantages over shallow reasoning and reasoning from first principles. Broad coverage of problems can be achieved through these strategies without a complete causal model or a complete set of pattern $\rightarrow$ action rules. This greatly reduces the initial knowledge acquisition effort. In addition, a large part of the knowledge base used in this type of reasoning may be readily used in solving similar tasks in a domain. The use of more complete models of components in the system improves the efficiency of the strategies.

The general control strategy, difference pursuit, reduces complexity by focusing on the most relevant portion of the system. The original focus is provided by general functional information and tests. When an observable symptom is detected, local functional information provides additional constraints on relevant components. When the point of appearance of symptoms is found, physical information is used to determine possible fault candidates. Local testing then distinguishes among the possible alternatives.

Specialized strategies may be combined with shallow or recognition-based reasoning in a multi-level system. Both efficiency and coverage are important in an expert system. For common or known faults, a precompiled sequence of operations may be retrieved and used. Specialized strategies may be used to provide coverage of novel or less common faults. Because the exploration phase can discover useful areas for detailed examination, the location of any symptoms obtained in this phase may be remembered to guide future diagnoses. In this manner, a fast and efficient diagnostic system can be achieved through the compilation of novel problems after they are solved.

8 Summary.

Specialized strategies and inexact models offer an alternative level of reasoning for expert systems. They provide extended coverage without specification of a complete causal model or extensive knowledge base acquisition. This approach appears particularly useful for diagnosis of complex, changing, or short-lived systems.

Acknowledgements.

Valuable direction and comments were provided by Paul Johnson and William Thompson. Imran Zualkernan contributed some initial ideas about diagnostic strategies. Thanks to John Carlis, Michael Wick, and two anonymous reviewers for comments. Thanks also to the technicians at IBM and CDC who helped us begin to understand computer hardware diagnosis.

References

[Davis, 1984] Randall Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.

[de Kleer and Williams, 1987] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[Genesereth, 1984] Michael R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.

[Gorry and Barnett, 1968] G. Anthony Gorry and G. Octo Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1:490–507, 1968.

[Hart, 1982] Peter E. Hart. Directions for AI in the eighties. *SIGART Newsletter*, (79):11–16, Jan. 1982.

[Johnson *et al.*, 1987] Paul E. Johnson, Imran Zualkernan, and Sharon Garber. Specification of expertise. *International Journal of Man-Machine Studies*, 26:161–181, 1987.

[Reiter, 1987] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[Thompson *et al.*, 1983] William B. Thompson, Paul E. Johnson, and James B. Moen. Recognition-based diagnostic reasoning. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, pp. 236–238, Karlsruhe, West Germany, August 8-12, 1983.