Fault Diagnosis of Electronic Systems Using Intelligent Techniques: A Review

William G. Fenton, *Member, IEEE*, T. M. McGinnity, *Member, IEEE*, and Liam P. Maguire

***Abstract*—In an increasingly competitive marketplace system complexity continues to grow, but time-to-market and lifecycle are reducing. The purpose of fault diagnosis is the isolation of faults on defective systems, a task requiring a high skill set. This has driven the need for automated diagnostic tools. Over the last two decades, automated diagnosis has been an active research area, but the industrial acceptance of these techniques, particularly in cost-sensitive areas, has not been high. This paper reviews this research, primarily covering rule-based, model-based, and case-based approaches and applications. Future research directions are finally examined, with a concentration on issues, which may lead to a greater acceptance of automated diagnosis.**

***Index Terms*—Artificial intelligence, case-based reasoning, fault diagnosis, fuzzy logic, machine learning, model-based reasoning, neural networks, rule-based reasoning.**

I. Introduction

WITH increasing system complexity, shorter product lifecycles, lower production costs, and changing technologies, the need for intelligent tools for all stages of product lifecycle is becoming increasingly important.

In [71], a system is defined as "any aggregation of related elements that together form an entity of sufficient complexity for which it is impractical to treat all of the elements at the lowest level of detail." Examples are automobiles, computers, or electronic circuit boards built using very large scale integrated (VLSI) components.

Fault diagnosis isolates the source(s) of a system malfunction, by collecting and analyzing information on system status using measurements, tests, and other information sources (e.g., observed symptoms). Often, it is performed by a human diagnostician, and it is an important function at all stages of the product lifecycle, but particularly during manufacture and field maintenance.

Over the last three decades, automating fault diagnosis using artificial intelligence (AI) techniques has been a major research topic. There has been much progress, but industrial acceptance, particularly in cost sensitive areas, has not been high.

In addition, with the emerging use of reconfigurable systems, in-line testing [58] and intelligent diagnostics can assist in the self-maintenance of complex systems [69].

The purpose of this paper is to review system diagnosis research using AI approaches, and to examine its application, particularly in the area of electronic system diagnosis.

II. Motivation

Fault diagnosis using intelligent techniques has been an active research area for the last two to three decades. However, its industrial acceptance has been low. Why? This paper forms the review section of the authors' investigation into this question. Additional work is on-going which will identify acceptance issues, and possible solutions to increased acceptance and deployment.

Furthermore, no single reference could be located, which covered a broad selection of approaches to intelligent diagnosis, so it was felt that this paper could fill this void.

III. Diagnostic Process

The purpose of fault diagnosis is to isolate the cause (component or subassembly) of a system malfunction in a timely manner.

The sequence followed to perform diagnosis can generally be summarized as follows.

1) *Fault Information Generation*: Information must be gathered about the nature of fault. This is achieved by fusing information from various sources including: observed symptoms, taking measurements, and running diagnostic tests.
2) *Fault Hypotheses Generation*: The information gathered is then used to localize the fault to a subset of components or subassemblies which are consistent with the available fault information.
3) *Fault Hypothesis Discrimination*: If more than one fault candidate is proposed it may be necessary to perform further tests or employ historical data (e.g., probabilities), to discriminate further. If further discrimination is not possible, experience or trial and error may be called on to determine the most appropriate repair.

Essentially the diagnostic process can be defined as fault isolation using information collected from system observations and tests.

Manuscript received November 30, 1999; revised January 4, 2001. This paper was recommended by Associate Editor P. K. Willett.

W. G. Fenton is with the Intelligent Systems Engineering Laboratory, School of Computing and Intelligent Systems, Faculty of Informatics, University of Ulster, Derry, U.K. He is also with International Test Technologies, Ltd., Letterkenny, U.K. (e-mail: billy_fenton@intertesttech.com.).

T. M. McGinnity and L. P. Maguire are with the Intelligent Systems Engineering Laboratory, School of Computing and Intelligent Systems, Faculty of Informatics, University of Ulster, Derry, U.K.

Publisher Item Identifier S 1094-6977(01)08911-8.

IV. Traditional Approaches

A. Rule-Based Systems

1) Approach: Rule-based diagnostic systems represent the experience of skilled diagnosticians in the form of rules which generally take the form "IF symptom(s) THEN fault(s)." Representing the knowledge for a particular problem domain, may require hundreds, or even thousands of rules. Rule-based inference involves taking information about the problem domain, and invoking rules which match this information. This generates new data which is added to the problem information. This process is repeated iteratively until a solution to the problem is found [50], [62]. Most intelligent diagnostic programs implemented in the 1970s and early 1980s were of this form.

2) Applications: A survey of a selection of applications in electronic engineering is described in [63]. Included are applications in the diagnosis of telephone networks, disk drives, telephone switching equipment, and avionics control systems.

Even more recently, rule-based systems are continuing to be used. An Expert System for PC Repair and Maintenance (ESPCRM) [47] describes a system for diagnosing PC systems to the replaceable module level. In [64], a program for diagnosing electronic forge press faults is reported. In [2], a complex expert system employing multiple specialized rulebases for diagnosing complex PC boards is described. Finally, in [21], a diagnostic tool for server computers boards which uses a rulebase to analyze a dump of the processor's internal memory is reported.

3) Issues: The primary advantage of this approach is its intuitive simplicity.

Its disadvantages are the following.

a) The difficulty of acquiring the knowledge to build the rulebase—known as the knowledge acquisition bottleneck.
b) Its ability to deal with novel faults.
c) System dependence, that is, a new rulebase will have to be generated for each new system type.

B. Fault (Decision) Trees

1) Approach: Historically, this has been the most commonly used method for documenting fault diagnosis procedures. A fault tree uses symptom(s) or test results as its starting point, followed by a branching decision tree, consisting of actions, decisions, and finally repair recommendations. Fig. 1 shows a simple example.

2) Applications: To assist in the navigation of large diagnostic networks, [1] describes a hypermedia system for a point-and-click traversal of fault trees and other types of diagnostic information.

To simplify the generation of fault trees for complex systems, intelligent techniques have been applied to automatically generate them. In [23], automatic fault tree generation is performed by using a circuit description, fault simulation to produce the electrical effects caused by failures, quantification and classification of these effects to produce a test matrix, and finally production of the test tree by recursively searching and evaluating the test matrix. In [57], fault trees are generated using cases extracted from a case-based reasoning system. In [36], process models, fault simulation, and machine learning techniques are applied to generate fault trees.

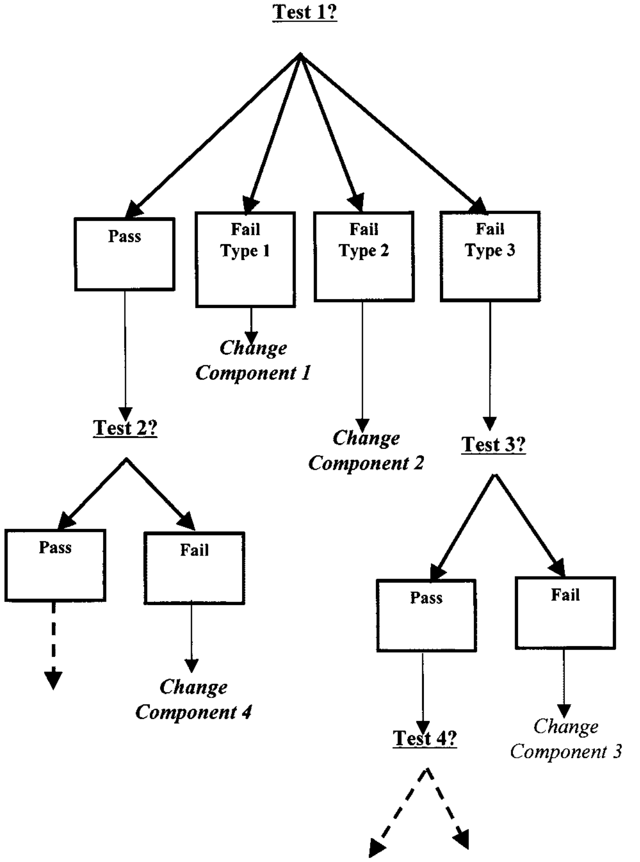


Fig. 1. Simple fault tree.

Also, fault trees have been used in various real-world intelligent applications including [23] which presents a system for diagnosing automotive electronic control systems and [66] which describes an expert system for color TV diagnosis.

3) Issues: The primary advantage of fault trees is simplicity and ease of use. In fact, little training is needed to use these diagnostic aids. However, for more complex systems, a full fault tree can be very large. In addition, a fault tree is system dependent and even small engineering changes can mean significant updates. Lastly, a fault tree offers no indication of the knowledge used to generate the answer.

V. Model-Based Approaches

Over the last 15 years, models have superseded rule-based techniques, as one of the premier research directions for intelligent systems diagnosis. A model is an approximate representation of the actual system being diagnosed. Model-based diagnosis involves using the model to predict faults using observations and information from the real device or system.

Models are often used in a hierarchical fashion, that is initial diagnosis is performed to a subunit level using a high level model and then a more detailed model of the subunit is used to diagnose to the next level and so on.

Various types of approaches have been used including fault models, structural models, behavioral models, and diagnostic inference models. The following sections discuss these various approaches, and their associated inference mechanisms.

	TP#1	TP#2	TP#3	TP#4	TP#5	TP#6	TP#7
Gate 1 Stuck at 0	X						
Gate 1 Stuck at 1					X		
Gate 2 Stuck at 0		X	X				X
Gate 2 Stuck at 1				X		X	
Gate 3 Stuck at 0							
Gate 3 Stuck at 1				X	X		X
Gate 4 Stuck at 0		X					
Gate 4 Stuck at 1			X		X		

Fig. 2. Simple fault model for a digital circuit.

A. Fault Models (or Fault Dictionaries)

1) Approach: This type of model anticipates the types of faults that may occur and only models these. Each of the selected fault types is inserted into each component, and using simulation, the behavior of the overall system is monitored. Each simulation produces a description of how the overall system operates when a particular part is defective in a specific way. This provides a list of fault/symptom pairs, which is used to produce a fault dictionary, which can indicate which component is defective when a particular overall symptom is present.

2) Applications: This method has primarily been applied to the diagnosis of digital circuits, where it has been used for the detection of stuck at "one" and "zero" faults, bridging (shorts) faults, and delay (timing) faults [80]. For example, to test a simple digital combinational circuit a series of binary test vectors is used. Using the fault simulator the behavior of each test pattern is noted for each fault type. Fig. 2 illustrates an example of a fault dictionary (x indicates that this test pattern fails if the corresponding fault is present). The defective component is the gate which behaves according to the fault dictionary or model for each test pattern.

3) Issues: For combinational digital circuits, fault models can diagnose modeled faults accurately, however, they are unable to deal with unanticipated (i.e., unsimulated) faults. However, the set of simulated faults may be adequate for most diagnostic purposes and, therefore, may provide a more than adequate solution for many applications.

Fault models are less successful when used with sequential circuits. To diagnose such circuits a test sequence rather than a single vector is required, and if the state of the circuit is lost during test because of a fault, it may not be possible to complete the sequence and, therefore, the diagnosis [80], [81]. Splitting the circuit into more manageable chunks, known as encapsulation, has been proposed as a possible solution [81].

Finally, for large circuits, the quantity of test vectors required can be large leading to impractical test times. Data compression approaches have been applied to this problem [80], [82].

B. Causal Models

1) Approach: A causal model is a directed graph where the nodes represent the variables of the modeled system and the links represent the relationships or associations between the variables. For example, in a diagnostic model, the variables often represent the symptoms and the faults, and the links represent the symptom–fault associations. The strength of each link is often defined using a numerical weight or probability. Therefore, the faults hypotheses formed are ranked or eliminated using Bayesian techniques [59], [60]. Bayesian networks are a variation on this approach [37].

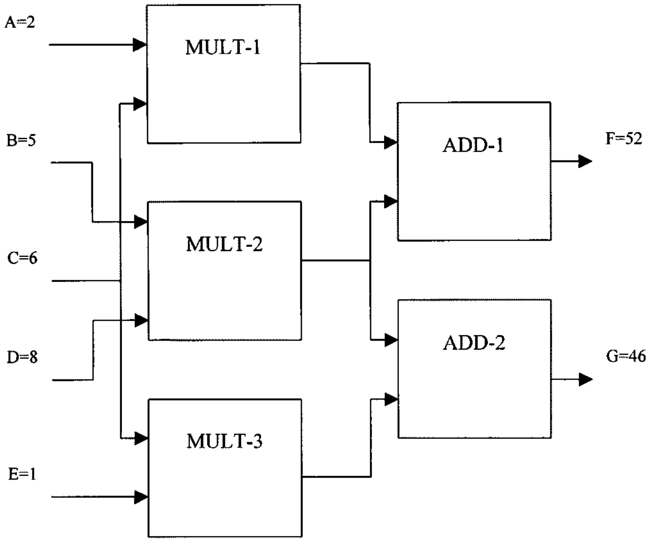


Fig. 3. Simple circuit model.

2) Applications: In [83], a Bayesian network is applied to the diagnosis of an integrated circuit tester. The knowledge of a domain expert regarding the probability of different tester failure modes is represented as a Bayesian network.

According to [84], rule-based systems are more prevalent than model-based approaches in industry because it is perceived that model-based systems are more difficult to build. To overcome this, a tool for converting a simple block diagram of a system to a causal model is presented.

3) Issues: Expert knowledge of the application area is needed to construct a causal model, so the "knowledge acquisition bottleneck" is its primary shortcoming.

The primary advantage is the ability to represent complex structured knowledge about physical or abstract concepts more easily than rules thus leading to greater computational efficiency. In addition, causal models are based on the firm mathematical theory of probability.

C. Models Based on Structure and Behavior

1) Approach: One of the primary research directions over the last 15 years has been the use of models based on structure and behavior. A dual representation of both structure and behavior is used. The structure representation lists all the components and their interconnections within the modeled system. The behavior representation describes the correct behavior pattern for each component. Behavior models can use various levels of abstraction including: mathematical, qualitative, or functional [10], [18]. Both representations are often created using logical formulae such as first order predicate calculus.

If the operation of the model does not agree with observations from the real system during a particular mode of operation, then a discrepancy has occurred and a diagnosis must be performed to find the defective component(s). Fig. 3 shows an example of a simple arithmetic circuit. If the inputs A through E are stimulated as shown, the outputs should measure as shown. Failure to measure these values indicates a discrepancy between the model and the real system.

Unlike fault models, this type of model is a correct model. That is, it models a working device, and theoretically, it can diagnose any fault type, not just the modeled ones.

Many of the basic techniques were proposed during the 1980s and involve the diagnosis of simple combinational digital circuits. The same basic principles apply to other device types. The process generally consists of three steps.

1) *Hypothesis Generation*: Generate a list of components (suspects) that might be responsible for the observed discrepancy.
2) *Hypothesis Testing*: Test each suspect to see if it can account for all the observations. A number of methods have been proposed for exonerating and reducing the list of suspects created during hypothesis generation. These include: constraint suspension [19], assumption-based truth maintenance [39], and fault model simulation [43].
3) *Hypothesis Discrimination*: If more than one suspect remains after the previous step, this action collects more information to aid in further discrimination. Additional information to discriminate can be collected using one of the following methods.

— *Additional Measurements*: Determining the best measurement sequence is the key issue and various approaches have been investigated including the use of failure probabilities [39] and the half-split method [43].

— *Additional Tests*: Determining the next test, which will provide the best information for maximum discrimination, is the key issue [68].

2) Applications: Hypothesis testing (HT) was one of the seminal works in structural/behavioral models for diagnostic applications [19]. Its application area was combinational digital circuits. To describe structure it used a subset of DECmmp Parallel Language, a VLSI design language. The representations used were hierarchical and both physical and functional descriptions were employed. Constraints were used to describe behavior, and both simulation and inference rules were used to describe the relationships between component inputs and outputs. Diagnosis was performed using candidate generation and constraint suspension.

In [28], HT is extended to deal with time variant digital circuits. Its behavioral representations are extended to deal with (value, time) pairs so that the behavior of a circuit can be described over a series of time periods. However, it concluded that unless complete state visibility (i.e., measurements being made at the end of different time periods) is available, diagnosis generation is inherently under-constrained and indiscriminate. Single stepping the circuit, so that observations could be taken at different time points, was proposed as a possible solution.

In [39], the general diagnostic engine (GDE) is introduced. GDE addressed the issue of multiple faults and became the basis for much later research in the area [39]. It introduced the use of ATMS [38] for diagnosis. Using constraint propagation and ATMS, it identifies minimal diagnoses, but considered all supersets of each minimal set a possible diagnosis; if a particular minimal diagnosis is exonerated all its supersets are also exonerated. To further discriminate amongst candidate diagnoses, it uses additional circuit measurements. To make an optimum set of measurements, it uses one-step look-ahead based on minimum entropy to predict the best probing sequence. Failure probabilities of individual components are needed to guide this process. Often such failure probabilities can be difficult to obtain, so an extension to GDE [41], proposes the use of crude probability estimates to guide diagnosis. It does this by assuming all components fail with equal probability and with extremely small probability. Lastly, some extensions to GDE exploit fault modes or models to provide additional diagnostic discrimination [40].

In [29], extended diagnostic engine (XDE) extends the GDE program described above, to deal with more complex circuits, including sequential ones. It uses a structural language called BASIL to provide both a physical and a functional representation of a circuit. For example, the functional representation would describe an arithmetic circuit in terms of adders and multipliers, whereas the physical description would describe the actual components used to build the circuit. The relationship between both descriptions is defined. To describe behavior, a temporal constraint propagation language called TINT is used. TINT defines rules at multiple levels of temporal abstraction to describe the operation of the circuit primarily at the functional level. Probability estimates are used to rank alternative diagnoses and to choose the next best measurement. These estimates are defined relative to a components complexity. To further refine diagnoses, fault models are employed to further adjust probability estimates. XDE has been tested on complex boards including microprocessor-based circuits.

In [15] and [17] DEDALE, an approach for analog circuit fault diagnosis, is described. DEDALE is an ATMS-like system. Components behavior is described using qualitative models based on relative orders of magnitude. Some components, such as transistors, can have a number of correct modes of operation. Diagnosis is performed in a hierarchal fashion, starting at the device level, which is diagnosed in a functional manner, and working down to the component level, block by block. To perform inference within a defective block, the measurement at each node and its attached components are checked for consistency with a correct model for the observed measurements. An inconsistency in the behavior indicates that one of the components attached to that node is defective. Intersection with other inconsistent nodes can further isolate the defective component.

CATS is a domain independent diagnosis engine based on the GDE framework, but with extensions to process values which are imprecise and change with time. DIANA is an implementation of CATS for diagnosing analog circuits [16], [17]. To allow for measurement imprecision, quantities are represented in CATS/DIANA using ranges or numeric intervals. Continuous signals are represented by using arrays of numeric intervals, accompanied by a triplet defining sample start instant, sampling increment, and number of samples. In order to repeat measurements the sample start instant must be synchronized in some way (e.g., clock signal). The imprecision of component parameters is also represented using numeric intervals. Component models are qualitative approximations, not suitable for accurate simulations, but adequate for troubleshooting purposes. The diagnostic engine CATS receives as input constraints from the models and measurements. Then using an ATMS-like inference mechanism it produces diagnostic candidates as outputs.

In [13], a generic model-based diagnostic system for a particular area of technical diagnosis is presented (switch-mode power supplies). A structural model based on frames, and a behavioral model based on heuristic rules which represent fault behavior in modules or components is used.

3) Issues: Models based on structure and behavior would appear to represent an ideal solution for many diagnostic problems. Theoretically, because of the use of correct models, all faults can be diagnosed; CAD data can be used to automatically generate suitable models. However, in practice, there are a number of significant limitations.

1) It is computationally intensive for complex problems [24]. Focusing on the most probable failures first [42] and the inclusion of fault models have been used to improve efficiency [40].
2) Representing the behavior of complex components, such as a Pentium microprocessor, is still a major research issue [20].
3) Complete and consistent models are hard to develop. Essentially, a model is only an approximate representation of a real world system. For example, a circuit bridging fault will not be represented in the structural model [20], [24].
4) Information relating to the ways the system can fail is often not present. This can lead to the isolation nonsensical faults [85].
5) Unless CAD generation is possible models can be time consuming to develop and maintain.

D. Diagnostic Inference Model

1) Approach: The diagnostic inference model [71], [72], performs diagnosis by representing the problem to be solved via the flow of diagnostic information. Previously known as the information flow model, the name change reflects the models focus on information provided by diagnostics and inferences that can be drawn from this information.

The model consists of two basic elements: tests and conclusions. Tests consist of any source of diagnostic information including, observable symptoms, logistics history, and results from diagnostic tests. Conclusions typically represent faults or units to replace. The dependency relationship between tests and conclusions is represented using a directed graph. In addition to tests and conclusions, there are three other possible elements in a diagnostic inference model: testable input, untestable input, and No-Fault. An input represents information entering the system which may affect the health of the system. A testable input can be examined for validity, an untestable input cannot. A No-Fault is a special conclusion indicating that the test set found no fault.

Fig. 4 shows an example of a diagnostic inference model.

Test sequencing is optimized using algorithms based on maximum test information gain. Diagnostic inference combines information from multiple tests using several logical and statistical inference techniques, including a modified form of Dempster–Shafer (D–S) evidential reasoning [50] which incorporates a special conclusion, the unanticipated result. The unanticipated result compensates for disappearing uncertainty in the face of conflict.

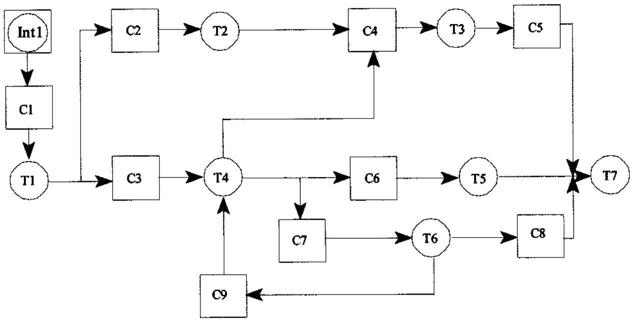


Fig. 4. Information flow model example.

As with all model-based techniques conflicting diagnoses may be derived. Conflicts are caused by: test error, multiple faults, or incomplete or inaccurate models. The D–S method and certainty factors are both used as methods for reasoning with these uncertainties [72].

2) Applications: Various successful uses of the diagnostic inference model are summarized in [86]. In [87], its application to radar system maintenance is outlined, and in [88] its application to the diagnosis of power supplies is described.

In [61], an approach similar to the diagnostic inference model is proposed and deployed for the troubleshooting of complex PC boards to component level. Models of the tests, rather than structure or behavior are used. The test models are specified in terms of how the tests act on the device under test (e.g., does the test access memory, output to port), and each test is mapped to specific components. This is combined with information on the degree to which each component is exercised, to give a relative weighting to each diagnosis using Bayesian-like probabilistic formula. The system now forms part of the Hewlett-Packard Fault Detective product.

3) Issues: Diagnostic inference models are at their most effective if considered and implemented at the design phase of the product lifecycle. Unfortunately with many systems, design for diagnosis is still not an important consideration, so an inadequate supply of structured diagnostic information makes accurate diagnosis difficult using this approach. However, if an adequate model can be built using available diagnostic information diagnosis can be both accurate and computationally efficient [86].

VI. Machine Learning Approaches

The approaches discussed in the previous sections, once implemented, will have a fixed level of performance. It is not possible to improve performance, by using the experiences of past successes and failures. Machine learning approaches exploit knowledge of previous successful or failed diagnoses to continually improve system performance or use available domain data to automatically generate knowledge.

A. Case-Based Reasoning

1) Approach: Case-based reasoning (CBR) involves storing experiences of past solutions known as cases, retrieving a suitable case to use in a new problem situation, adapting and reusing the retrieved case to suit the new problem, revising the adapted case based on it's level of success or failure, and eventually retaining any useful learned experiences in the case memory [45], [46].

A CBR solution generally consists of the following steps.

- Knowledge or Case Representation.
- Case Retrieval.
- Case Reuse.
- Case Revision.
- Case Retainment (or learning).

Case representation, often called case memory, consists of deciding what to store in a case, selecting an appropriate structure for representing the case contents, and deciding on a suitable case indexing scheme to enable efficient retrieval.

Case retrieval consists of the following steps.

1) Identify features which sum up the current problem or case.
2) Use the features to find similar cases in the case memory. These are ranked in order of similarity.
3) Perform final matching by analyzing in more detail the cases selected in step 2) against the current case. Select the most similar case.

Case reuse consists of finding the differences between the past and the current case, and then adapting the past case in some way to match the current case. Common forms of adaptation include substitution (substituting new values for old) and transformation (using heuristics).

Case revision involves evaluating the case solution from the reuse phase, and if necessary repairing any parts of the solution which are contributing to an inadequate solution. Evaluating involves applying the solution in a real situation, and measuring in some way its level of success. Errors in the solution are then detected and repaired using domain specific knowledge.

Finally, case retainment (or learning) adds useful information learned during the current problem solving task to the case memory. This may not only be successful new cases, but also failed cases ("don't do that again!"). Retainment can be an adjustment to an existing case(s) and its indeces or the addition of an entirely new case.

2) Applications: In [73], each case is represented using an ID number, frequency, symptoms, and actions. On retrieval it uses a possibility metric to rank cases; this is based on similarity and frequency. The problem of generating casebases for new products is discussed, and the solution of using two casebases, a generic casebase, and a product-type casebase is proposed. The generic casebase stores domain diagnostic rules based on symptom-defect causalities. The product-type casebase is generated from the generic casebase by specializing its cases and updating the frequencies.

In [14], an incremental case-based electronic fault diagnosis system is presented. A minimal case description can be used to perform initial case retrieval. The retrieved set is examined to determine tests which the operator is asked to perform and these results are used to discriminate between cases.

In [5], a circuit diagnosis support system for electronic assembly operations is described. Real-time diagnosis is required, so CBR is chosen over model-based diagnosis (MBD), as the computational overhead of MBD is considered to be too high. Initial case retrieval is performed, and additional tests are optimally selected, using dynamic programming techniques or heuristics, to refine the diagnosis. The case-base is updated after each diagnosis to reflect previously unknown faults. After five weeks of on-line use the system could diagnose 95% of defects.

3) Issues: The effectiveness of CBR depends on the availability of suitable case data, generated from historical data or simulation, and the selection of effective indexing, retrieval, and adaptation methods.

B. Explanation-Based Learning

1) Approach: Explanation-based learning (EBL) uses domain knowledge, and a single training example, to learn a new concept [50]. For example, in diagnosis, a system model and an example of misdiagnosis can be used to derive an explanation of an appropriate diagnosis.

2) Applications: In [67], a diagnostic EBL system is described which improves diagnostic inference models following learning. It operates as follows. After a misdiagnosis, further testing is performed until a correct diagnosis is made; this additional knowledge is then used to modify the model so that the correct diagnosis is consistent with the testing.

3) Issues: EBL success depends on the availability of adequate domain knowledge. Therefore, for complex domains where extensive knowledge is needed to formulate new concepts, the approach may prove to be intractable.

C. Learning Knowledge From Data

1) Approach: Another approach is the extraction of knowledge bases from existing databases or casebases. This overcomes the knowledge acquisition bottleneck and automatically generates an intelligent diagnostic system from existing resources. Obviously, it is only useful if prior data is available, so for new systems it is of little or no use.

2) Applications: In [76], knowledge base generation from General Motor's diagnostic database is described. This database contains 300 000 cases of vehicle symptoms and repair information. An extended form of the decision tree induction algorithm ID3 [50] is used to extract general diagnostic rules from the database. ID3 uses database examples to generate decision trees, which are then used to classify the examples into suitable diagnostic rules. The extensions deal with the presence of inconclusive data sets, that is, when the set of examples used is not enough to specify a single conclusive outcome.

3) Issues: Using existing information to automatically generate a knowledge-based system can greatly speed development time and greatly reduces the knowledge acquisition bottleneck. However, it is only suitable where large databases of domain information are available. Therefore, it is inappropriate for new systems where actual data is not yet available.

VII. Other Approaches

A. Fuzzy Logic

1) Approach: Fuzzy logic provides mechanisms to represent and manipulate linguistic concepts such as the following.

— The water is very hot.

— The signal on the oscilloscope is a bit noisy.

It deals with approximates rather than exact measurements and is based on fuzzy set theory [12], [77], [78], [79].

In traditional sets, membership is either true [1] or false [0], and there is no concept of partial membership. In fuzzy sets, partial membership is allowed, so membership is represented by a value between 0 (definitely not a member) and 1 (definitely a member). In fuzzy set theory, a series of operators is defined, for manipulating sets. Many of these are analogous to those used in conventional sets, such as, union (OR), intersection (AND), and complement (NOT). Fuzzy reasoning consists of manipulating a series of unconditional and conditional fuzzy propositions or rules using fuzzy rules of inference.

With its concept of partial set membership, fuzzy logic provides a good alternative for reasoning with uncertain and inaccurate data.

2) Applications: Most of the research work relating to fuzzy logic and diagnosis has occurred in the area of dynamic industrial processes. In this domain, fuzzy logic has been applied primarily to the following tasks [26], [32], [65].

a) Fault Detection: Industrial processes are characterized by dynamic continuous variables (symptoms). Such variables are prone to measurement errors, noise, and operating conditions. Therefore, reliable measurement thresholds are difficult to define. Fuzzy logic provides a good solution to this problem, by representing signal values using overlapping linguistic variables.

b) Fault Diagnosis: Fault diagnosis in dynamic processes is always approximate, as measured signal values are only known to a certain degree of accuracy. A fuzzy inference system based on fuzzy IF–THEN rules can provide a solution to this problem, and is proposed and reported by many researchers.

Applications, which apply fuzzy logic to the diagnosis of electronic systems are also reported. FLAMES is a program for troubleshooting analog circuits [53], [54]. It is a GDE-like program employing ATMS. However, continuous signals and component parameters are represented using fuzzy values, and fuzzy values can be propagated across the circuit model. In [8], a similar system using possibility theory (a form of fuzzy logic) [22], to improve the accuracy of diagnosis of analog circuits is reported. In [69], a practical application, which uses fuzzy qualitative values for sensor measurements, in the development of self-maintenance photocopiers, is presented.

3) Issues: Because of it, use of linguistic variables' fuzzy logic provides a very human-like and intuitive way of representing and reasoning with incomplete and inaccurate information. It is typically combined with other approaches such as rules, models or cases, and provides a good alternative for reasoning under uncertainty.

B. Artificial Neural Networks

1) Approach: The human brain is constructed of billions of interconnected cells or miniprocessors called neurons. Artificial neural networks (ANN) are inspired by the brain's neural circuitry and use the approach for complex problem solving [33].

ANNs can be considered as weighted directed graphs, the neurons being the nodes, and the connections between the nodes being weighted links. Groups of nodes are arranged in layers. There are two basic network architectures, and many variations of each type.

— Feed-Forward: No feedback between layers. Previous inputs are not remembered; only the current input is operated upon. Examples are multilayer perceptrons and radial basis function nets.

— Recurrent: Feedback between layers. These networks remember prior inputs so they can be taught to perform a sequence of steps or to construct associative memory. Examples are artificial resonance theory (ART) models, Hopfield nets, Kohonen SOM, and competitive nets.

Before operation, ANNs must be trained using training examples. Using the examples, the network weights and threshold functions are adjusted using a suitable learning algorithm. There are two categories of learning.

— Supervised Learning: For each training example a correct output is provided. Weights are then adjusted until the actual output is as close as possible to the correct output. Back propagation is an example of a supervised learning algorithm.

— Unsupervised Learning: In this case a correct output is not provided for each training example. The training examples are used to explore the underlying structure of data. The associative memory learning algorithm for Hopfield Nets is an example.

2) Applications: In [56], an application of ANNs in the diagnosis of simple combinational digital circuits is described. A multilayer feedforward network is trained using back-propagation and is designed to detect single faults in a one bit full adder. The inputs consist of circuit inputs and outputs, and internal test points. The outputs represent the defective component and fault type (none, stuck-open, stuck-closed). It operated well on single faults. The authors checked its ability to generalize by testing it with multiple faults; it did not generalize well.

In [35], a multilayer perceptron trained using back-propagation is used for diagnosing digital circuits. The input layers accept the pass/fail status of a test vector set and the output layer equals single faults. Trained with a fault dictionary for single faults it showed 100% success for single fault diagnosis and 75%+ success with two faults.

In [74], the diagnosis of telephone exchange line cards using ANNs at British Telecom is described. The authors had already explored and implemented a model-based approach to the same problem, but they wished to investigate the use of an ANN trained with historical data to achieve the same task. It was felt that an ANN solution could be implemented much more rapidly if the historical data was available. A three-layer feedforward network, with circuit measurements as the inputs, and component pass/fail as the outputs, was constructed and trained using back-propagation. Comparing their experiences with model-based approaches, the authors summed up as follows. ANNs can be trained directly from data, are good with common faults, and provide rapid diagnosis. MBD can diagnose obscure faults, provides graphical support, and can explain a diagnosis.

In [51], an application for the diagnosis of multiple faults using multilayer perceptrons (MLP) is presented. The circuit is a bipolar section of an analog IC. It is stimulated using a sine wave and the magnitude of the Fourier harmonics in the spectrum of the circuit output is measured to verify and diagnose the circuit. A signature representing the output measurement is the input to the MLP, and the outputs represented the location and resistances (types of faults corresponds to technological problems of dopage of the extrinsic zones or an open contact problem before metallization) of the faults. The MLP was trained using back-propagation, to detect single, dual, and triple faults using data generated via simulation.

In [34], an ANN is designed which assists a technician in circuit diagnosis (e.g., next best node to measure). A three-layer network is used; inputs are either on or off, and represent symptom states, pins observed to be good, pins observed to be bad, and a flag indicating whether the overall circuit is good or bad; the outputs indicate the next best points to test. The input-hidden layer use an unsupervised learning paradigm to form self organizing feature maps containing knowledge about fault symptoms represented in topographical order. Once the feature maps are formed, the hidden-output layer is trained using a supervised learning paradigm based on the delta rule, to indicate the next best location to test.

As well as equipment diagnosis, ANNs have also been applied to the diagnosis of dynamic processes. In this area, ANNs have been used to process the outputs of sensors and to perform diagnoses by using symptom-fault networks [25], [26], [75].

In [71], an ANN is used to determine when enough evidence has been gathered to draw a diagnostic conclusion. The approach used, is to terminate when a pattern of certainty values indicate a conclusion can be drawn. A three-layer network with three inputs and one output is used. The inputs represent highest expected probability, second highest expected probability, and the probability of an unanticipated result. The activation level of the output determines whether or not to terminate. Training was via back-propagation and used training data collected from experts.

3) Issues: The power of ANNs is their ability to approximate and recognize patterns. In diagnostic applications they have shown great promise in areas where noise and error is present. The diagnosis of analog circuits is an example. However, their scalability to large systems and circuits is questionable, and they may best be used to assist other techniques in dealing with error and noise.

VIII. Hybrid Approaches

Hybrid solutions combine techniques to produce improved diagnostic solutions by capitalizing on the advantages of the individual techniques.

A. Model-Based Reasoning and Case-Based Reasoning

A system which uses CBR to improve the MBD process is described in [24]. Device models consist of a structural decomposition of the device, with the device at the root of the tree, modules at intermediate nodes, and the replaceable components at the leaves. Each component and submodule has a failure pattern associated with it stored in a database. This failure pattern is a combination of sensor outputs. Diagnosis consists of traversing the hierarchy based upon observations, and the output is a ranked list of diagnoses. CBR is used to further refine the MBD process. All past diagnostic scenarios are stored as cases, which are indexed from the bottom of the device structural decomposition. Both good and bad diagnoses are indexed. For bad diagnoses, the correct failure is also referenced from the stored case. Experimental results showed, that by incorporating CBR at the end of the MBD process, significant improvements in the number of correct diagnoses is achieved.

Another system which combines MBD and CBR is described in [7]. Again the motivation is to improve diagnoses because of the use of incomplete and inaccurate models. Models are represented using the following.

— A devices structural decomposition.
— A list of failure types for each component.
— A list of symptom types.
— The different tests that can be performed to narrow down diagnoses.
— Relationships between failures, symptoms, and tests. Weights can be assigned to these.

Cases consist of the following.

— Universal knowledge-base about equipment diagnoses.
— Specific device knowledge about a device.
— Historical cases.

MBD and CBR are employed in two ways.

— Uses CBR to refine the list of diagnoses produced by MBD.
— Uses recorded cases to improve the model.

This is done in a number of ways including the following.

— Updating the failure rate of device modules.
— Adjusting the weights that relate test results to modules.
— Discovering and updating a new fault mode.

In [70], a system is presented which extracts an information flow model from a historical casebase which contains lists of test outcomes and appropriate diagnoses.

B. Model-Based Reasoning (MBR) and Fuzzy Logic

A fuzzy extension to the diagnostic inference model is described in [6]. It uses fuzzy logic, in its front end to deal with the uncertainty of measurements, and internally to generate membership degrees for faults, predicted by the outcomes of multiple tests. In [8], [53], and [54], and extensions to the model-based ATMS architecture, employing fuzzy logic is described. Essentially, fuzzy logic is used to improve the accuracy of component modeling and measurements.

C. Case-Based Reasoning (CBR), Artificial Neural Networks, and Fuzzy Logic

In [48], a connectionist case-based diagnostic expert system which learns incrementally, is reported. Designed for Singapore Airlines to assist technicians in troubleshooting inertial navigation systems, it consists of two parts, a connectionist module, and a flowchart module. The connectionist module is a three-layer feedforward network taking symptoms as its inputs and producing component faults as its outputs. It is trained using historical cases. The flowchart module is invoked if the ANN fails to produce a diagnosis. This consults the technician, and a new case is constructed, which is inputted to the connectionist module as a new case to perform incremental learning.

In [49], a case-based diagnostic system using fuzzy neural networks is described. The system is used to diagnose telecommunications systems. Fuzzy rules relating symptoms to faults are encoded in the network architecture. The network is a three- layer feedforward network. The input data is fed through possibility measure nodes, then through fuzzy-AND neurons in the hidden layer, and finally through fuzzy-OR neurons in the output layer. It is trained using historical data from an existing helpdesk.

D. Model-Based Reasoning and Genetic Algorithms

Genetic algorithms (GAs) [4], [31] employ the concepts of genetic evolution (chromosomes, genes, crossover breeding, and mutation), to perform search, learning, and optimization. In [30], MBD is primarily used, but GAs are used to optimize the best sequence of measurements.

IX. Diagnostic Standards

The importance of artificial intelligence (AI) in test and diagnosis is emphasized by the recent publication of a set of IEEE standards [89] which address the use of AI systems in test and diagnostic environments. Known as AI-ESTATE (“Artificial Intelligence Exchange and Service Tie to All Test Environments”), there are two component standards. IEEE 1232.1 provides a standard representation of test and diagnostic data and knowledge and interfaces between reasoners and other functional elements of a test environment. IEEE 1232.2 defines the communications mechanisms and services between reasoners and other functional elements of a test environment.

The primary goal of AI-ESTATE is to provide a methodology for developing diagnostic systems that will be interoperable, have transportable software, and, therefore, move beyond vendor and product specific solutions.

X. Commentary

Three broad classes of knowledge have been applied to diagnosis—heuristic, fundamental, and historical [5]. Heuristic knowledge employs rules and/or procedures, which relate symptoms to faults, often with associated certainty values or probabilities. IF–THEN rules are an example. Fundamental knowledge uses the underlying physics of the device to reason from first principles. Model-based reasoning (MBR) is an example. Historical knowledge employs data or experiences recorded during previous diagnostic sessions to perform new diagnoses. CBR is an example. Each with its pros and cons is suited to different application domains, or hybrid solutions can be constructed, exploiting the combined pros and cons of each approach.

A. Rule-Based Approaches

As rule-based systems strive to encompass the knowledge of a domain expert, in the form of rules (often hundreds or thousands), development and maintenance can be complex and time consuming. Particularly, for systems with short lifecycles (many electronic systems), it may not be worth the development effort. In addition, only faults anticipated during the design phase can be diagnosed. Conversely, their intuitive simplicity makes rules easy to understand and the inference sequence used for a particular problem can easily be traced. Additionally, the technique is well proven, with many rule-based systems having been deployed in real applications [2], [21], [63].

B. Model-Based Approaches

Models based on structure and behavior would seem to offer the ideal solution for many diagnostic applications; theoretically the models can be generated from CAD, and all defects including multiple faults can be diagnosed, without prior knowledge of the defect. However, in practice, a number of major shortcomings have become evident.

1) Full MBR becomes computationally intractable on problems with large numbers of components [42]. Various solutions have been offered, including introducing fault modes, and focusing on the most likely defects first [40], [42].
2) Diagnosing complex devices with a large number of simple components (e.g., gates), although requiring much computing power, is feasible. However, finding suitable behavioral representations for more complex components (e.g., microprocessor) continues to be a serious research challenge [20].
3) It is hard to develop a complete and consistent model, which can consider all fault types. After all, a model is only an approximation of a real-world device. For example, how can a bridging fault be represented in a structural model? And the tradeoff between model completeness and speed of diagnosis must be considered, that is, a more complete model will deliver slower and more accurate diagnoses, and vice versa [20]. Essentially, diagnosis is only as good as the model.
4) Knowledge of fault types is often not included, and this can lead to the diagnosis of nonsensical faults [85].
5) Development times can be long if CAD data cannot be used.

The diagnostic inference model “models the information provided by a set of tests with respect to a set of desired conclusions” [72]. Again results are only as good as the model, and the model is only as good as the set of available tests. However, the approach has been successfully applied to many real applications [86]–[88].

A similar approach is taken in [61], where a system for diagnosing complex PC boards is described. Here, structure/behavior models are rejected as being too complex to develop, and models of diagnostic tests and their associated components are used instead. The system has been deployed and cost savings of 7 million French francs was made during year one.

In summary, models based on structure/behavior, which can be applied in real situations are difficult to develop for complex systems. Whereas, diagnostic inference models based on the many successful real applications, would seem to be more practical for complex real-world problems.

C. Case-Based Approaches

Case-based systems depend on past diagnostic experiences to perform new diagnoses. In practice, CBR has proved to be effective in real-world circuit diagnosis applications [5], [14]. Issues include the following.

1) The inability to diagnose until an adequate case-base becomes available. In [5], an application involving the diagnosis of consumer electronics products is described, and it is reported that the system could diagnose 90% of defects after six weeks of operation, however, the domain complexity is not apparent. Additionally, less common faults will be more difficult to diagnose due to their lack of presence in the initial casebase.
2) Compared to rule-based and model-based systems it is not always apparent how conclusions are arrived at [14], as the diagnosis is based on the overall fault pattern, rather than a logical sequence of steps.
3) In [14], development times and performance were reported to be better than an equivalent model-based solution previously reported in [13].
4) Development and maintenance is easier than for traditional solutions such as rules, as knowledge acquisition is on-line and incremental [5].
5) Efficiency may be hindered by the indexing and retrieval mechanisms used particularly as the case-base begins to grow [24].
6) How domain specific are CBR solutions? A human technician can apply troubleshooting techniques learned on one product to a different product, by extracting general purpose rules or procedures from specific experiences. Can this be applied to CBR? In [73], the casebase is divided into generic and specific knowledge. However, the generic casebase is defined by domain experts and is not incremental.

D. Fuzzy Logic and Neural Networks

Fuzzy logic has been used as an extension to other methods, such as rules and models, to deal with uncertainty and incompleteness. Neural networks have been applied to various diagnostic problems, but their ability to deal with complex domains is questionable. In practice, both will probably form useful additions to hybrid solutions for real-world applications domains.

E. Hybrid Approaches

A primary research direction has been the combined use of MBR and CBR in diagnostic systems. Models are often inconsistent and incomplete resulting in inaccurate diagnoses. In addition, operators can input inaccurate information again leading to inappropriate conclusions. Supplemented by cases, irrelevant conclusions can easily be pruned from a candidate list of diagnoses. In addition, the model can be updated and improved using case data [7], [24].

MBR can be too slow for real-time applications so CBR may be a better alternative [5]. However, to supplement the CBR approach off-line, models can be used to verify new cases created by the adaptation process, or models can be used to initialize a case-base for a new product. And, in [70], models are generated from available case data.

Fuzzy logic has been combined with model-based reasoning, particularly in the domain of analog circuit diagnosis. Circuit measurements are represented using fuzzy values, and inferences are propagated using fuzzy techniques [6], [8], [53], [54].

Finally, in [52] a proposed hybrid architecture employing models, cases, and fuzzy inference, for diagnosing microprocessor-based boards is described.

XI. Future Directions

As electronic systems increase in complexity, the need for automated diagnostic tools has become more acute. This is exacerbated by reduced time-to-market, and shorter product lifecycles, leading to little development time being available for diagnostics. Although much research has been carried out in the area, much remains to be done, particularly in the deployment of useful tools, which save dollars, in real applications. Without a return on investment, there will be no implementation and no deployment. Some issues for future research are briefly discussed in this section.

1) Most complex electronic systems are now microprocessor or digital signal processor (DSP) driven. Most research has concentrated on hardware-only systems which consist of inputs, circuitry, and outputs. Processor-based boards involve the tight integration of hardware and software, and, therefore, present additional problems including the following.
   — Software test programs are generally used to test the hardware, but often these cannot be started if the board is defective.
   — The test programs often only provide a pass/fail result. What is an alternative test architecture which includes diagnosis without increasing the cost of test generation? On a manufacturing line, diagnosis is currently performed off-line using expensive debug technicians because diagnosis will often slow down the rate of production and, therefore, increase costs.
2) As product lifecycles reduce, fast deployment is a key issue [61]. For example, many PC systems have a lifecycle of three months. Developing diagnostics models is time-consuming unless CAD data can be used [27]. Using cases suffers from the initial lack of suitable cases

and a three to six month lifecycle does not give enough time to overcome this.

3) In [84], it is claimed that rule-based approaches are prevalent in industry and that the deployment of model-based approaches has been delayed by the perception that model-based solutions require specialized knowledge to enable implementation. To overcome this, a tool, which converts simple block diagrams of a system to a causal model is presented. Clearly, tools which simplify the development of intelligent diagnostic solutions are required and these tools must cater to the needs of engineers who may have little knowledge of AI.
4) Models based on structure and behavior have problems when scaled up to large circuits [42]. Particularly, representing devices with complex behaviors (e.g., Pentium microprocessor) continues to be a problem [20]. A suitable ontology [11] or representation vocabulary is needed for the electronic system domain and with specific representations for particular device types. For example, in [13], a structure/behavior model-based solution with a representation vocabulary suited to the domain of switch-mode power supplies only, is deployed successfully. In comparison, the more generic MBR solutions have not been successfully applied to complex real-world circuit diagnosis to our knowledge.
5) Structure/behavior models use a correctly functioning model. What about defects which change the structure of the model (e.g., bridging fault) thus making the model incomplete [20]?
6) Hybrid solutions form a continuing area of investigation, particularly the combined use of models and cases. Models suffer from the complexity versus completeness issue. If too complex, diagnosis can become intractable. If incomplete, diagnosis can be rapid but inaccurate. Conversely, CBR only becomes accurate after a period of deployment. Therefore, cases can be used to supplement and improve the diagnosis of an incomplete model and models can be used to initialize and verify cases. However, what complexity of model supplemented by cases, will provide fast and accurate diagnosis from initial deployment, where no cases are available, but yet is simple enough to be developed within an acceptable timeframe [7], [24]?
7) Most CBR solutions only collect cases, which are relevant to a specific system type. A new product means starting all over again. Is it possible to collect generic cases or experiences? For example, a human technician can carry experiences learned on old products to new products. Can cases be stored in a more generalized way?
8) Collecting diagnostic information using probing, forms part of many past works on circuit diagnosis. However, with modern circuit boards, probing is becoming less of an option, as packaging densities increase. More information will have to be collected via diagnostic tests [44].
9) Design for test (DFT) has become more prominent as system test becomes more difficult. Can DFT strategies incorporate diagnosis without compromising test cost and quality?
10) Electronic systems diagnosis is an expensive activity requiring high skill. As part of manufacturing, it is performed off-line by debug technicians. Using automated techniques, can it be performed as part of an on-line test [5] or can it be performed off-line by operators [61]?

XII. Summary

Increasing costs, shorter product lifecycles, and rapid changes in technology are driving the need for automated diagnosis. Although research has been active over the last two decades, much remains to be done. Primarily, the developed techniques must be scaled up to deal with current and future technologies but with improved development times and costs. Otherwise, acceptance will be difficult, particularly in cost sensitive domains, such as PCs and consumer electronics. To date, there have been some applications, but the general use of intelligent diagnostic solutions for electronic system diagnosis has yet to happen.

Acknowledgment

The authors would like to acknowledge the comments and suggestions of the various reviewers which have been incorporated in this revised manuscript.

References

[1] S. Abu-Hakima, "Visualizing and understanding diagnoses," *Can. Artif. Intell.*, vol. 30, pp. 4–8, Autumn 1992.

[2] D. Allred *et al.*, "Agatha: An integrated expert system to test and diagnose complex personal computer boards," in *Proc. Innovative Applicat. Artif. Intell.*, 1991.

[3] N. S. C. Babu and V. C. Prasad, "Radial basis function networks for analog fault isolation," *J. Circuits, Syst., Comput.*, vol. 7, no. 6, pp. 643–655, 1997.

[4] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on history and current state," *IEEE Trans. Evolutionary Computat.*, vol. 1, pp. 3–17, Apr. 1997.

[5] A. Balakrishnan and T. Semmelbauer, "Circuit diagnosis support system for electronic assembly operations," *Dec. Support Syst.*, vol. 25, pp. 251–269, 1999.

[6] T. M. Bearse and M. L. Lynch, "Fault hypothesis computations using fuzzy logic," in *Research Perspectives and Case Studies in System Test and Diagnosis*, J. W. Sheppard and W. R. Simpson, Eds. Boston, MA: Kluwer, 1998.

[7] M. Ben-Bassat, I. Beniaminy, and D. Joseph, "Combining model-based and case-based expert systems," in *Research Perspectives and Case Studies in System Test and Diagnosis*, J. W. Sheppard and W. R. Simpson, Eds. Boston, MA: Kluwer, 1998.

[8] C. Bos-Plachez, "A possibilistic ATMS contribution to diagnose analog electronic circuits," *Int. J. Intell. Syst.*, vol. 12, pp. 849–864, 1997.

[9] M. Catelani and M. Gori, "On the application of neural networks to fault diagnosis of electronic analog circuits," *Measurement*, vol. 7, no. 2, pp. 73–80, 1996.

[10] B. Chandrasekaran, "Functional representation and causal processes," *Adv. Comput.*, vol. 38, pp. 73–143, 1994.

[11] B. Chandrasekaran, J. Josephson, and V. Benjamins, "What are ontologies and why do we need them?," *IEEE Intell. Syst.*, pp. 20–26, Jan./Feb. 1999.

[12] E. Cox, *The Fuzzy Systems Handbook*. New York: Academic, 1994.

[13] P. Cunningham, "A case study on the use of model-based systems for electronic fault diagnosis," *Artif. Intell. Eng.*, vol. 12, pp. 283–295, 1998.

[14] P. Cunningham, B. Smyth, and A. Bonzano, "An incremental retrieval mechanism for case-based electronic fault diagnosis," *Knowledge-Based Syst.*, vol. 11, pp. 239–248, 1998.

[15] P. Dague, O. Raiman, and P. Deves, "Troubleshooting: When modeling is the trouble," in *Proc. 6th Nat. Conf. Artif. Intell.*, 1987, pp. 600–605.

[16] P. Dague *et al.*, "Analog systems diagnosis," in *Proc. 9th Euro. Conf. Artif. Intell.*, 1990, pp. 173–178.

[17] P. Dague, "Model-based diagnosis of analog electronic circuits," *Ann. Math. Artif. Intell.*, vol. 11, pp. 439–492, 1994.

[18] ——, "Qualitative reasoning: A survey of techniques and applications," *AICOM*, vol. 8, no. 3–4, pp. 119–192, Sept./Dec. 1995.

[19] R. Davis, "Diagnostic reasoning based on structure and behavior," *Artif. Intell.*, vol. 24, pp. 347–410, 1984.

[20] R. Davis and W. Hamscher, "Model-based reasoning: Troubleshooting," in *Readings in Model-Based Diagnosis*, J. de Kleer, W. Hamscher, and L. Console, Eds. San Mateo, CA: Morgan Kaufmann, 1992.

[21] N. Dev and B. Anderson, "Pimtool: An expert system to troubleshoot computer hardware failures," *Proc. Innovative Applicat. Artif. Intell.*, 1997.

[22] D. Dubois and H. Prade, "An introduction to possibilistic and fuzzy logics," in *Non-Standard Logics for Automated Reasoning*, P. Smets *et al.*, Eds. New York: Academic, 1988, pp. 287–326.

[23] O. Duffaut and C. Castel, "A method to aid test-tree generation for hardware fault isolation in automotive electronic control systems," in *Proc. TOOLDIAG Int. Conf. Fault Diagnosis*, 1993, pp. 230–237.

[24] M. P. Feret and J. I. Glasgow, "Case-based reasoning in model-based diagnosis," *Applicat. Artif. Intell.*, vol. 7, pp. 679–692, 1992.

[25] P. M. Frank and B. Koppen-Seliger, "Fault detection and isolation in technical processes with neural networks," in *Proc. 34th Conf. Dec. Contr.*, 1995, pp. 2414–2419.

[26] ——, "New developments using AI in fault diagnosis," *Eng. Applicat. Artif. Intell.*, vol. 10, no. 1, pp. 3–14, 1997.

[27] M. R. Genesereth, "The use of design descriptions in automated diagnosis," *Artif. Intell. 24*, pp. 411–436, 1984.

[28] W. Hamscher and R. Davis, "Diagnosing circuits with state: An inherently underconstrained problem," in *Proc. 4th Nat. Conf. Artif. Intell.*, 1984, pp. 142–147.

[29] W. C. Hamscher, "Modeling digital circuits," *Artif. Intell.*, vol. 51, no. 1–3, pp. 223–271.

[30] B. Han, S.-J. Lee, and H.-T. Yang, "A model-based diagnosis system for identifying faulty components in digital circuits," *Appl. Intell.*, vol. 10, pp. 37–52, 1999.

[31] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[32] R. Isermann, "On fuzzy logic applications for automatic control, supervision, and fault diagnosis," *IEEE Syst., Man, Cybern. A*, vol. 28, pp. 221–235, Mar. 1998.

[33] A. K. Jain and J. Mao, "Artificial neural networks: A tutorial," *IEEE Computer*, pp. 31–44, Mar. 1996.

[34] O. Jakubowicz and S. Ramanujam, "A neural network model for the fault diagnosis of digital circuits," in *Proc. Int. Conf. Neural Networks*, vol. 2, 1989, pp. 611–614.

[35] A. A. AL-Jumah and T. Arslan, "Artificial neural network based multiple fault diagnosis in digital circuits," in *Proc. ICCAS*, vol. 2, 1998, pp. 304–307.

[36] D. Juricic, A. Znidarsic, and D. Fussel, "Generation of diagnostic trees by means of simplified process models and machine learning," *Eng. Applicat. Artif. Intell.*, vol. 10, no. 1, pp. 15–29, 1997.

[37] H. Kirsch, "Bayesian nets—A tool which makes Bayes Rule useful for diagnosis," in *Proc. Int. Conf. Fault Diagnosis (TOOLDIAG)*, 1993, pp. 748–756.

[38] J. de Kleer, "An assumption-based TMS," *Artif. Intell.*, vol. 28, pp. 127–162, 1986.

[39] J. de Kleer and B. C. Williams, "Multiple fault diagnosis," *Artif. Intell.*, vol. 32, pp. 163–196, 1987.

[40] ——, "Diagnosis with behavioral modes," in *Proc. IJCAI*, 1989, pp. 104–109.

[41] J. de Kleer, "Using crude probability measures to guide diagnosis," *Artif. Intell.*, vol. 45, pp. 381–391, 1990.

[42] ——, "Focusing on probable diagnosis," in *Proc. Nat. Conf. Artif. Intell.*, 1990, pp. 842–848.

[43] J. de Kleer and J. S. Brown, "Model-based diagnosis in SOPHIE III," in *Readings in Model-Based Diagnosis*, J. de Kleer, W. Hamscher, and L. Console, Eds. San Mateo, CA: Morgan Kaufmann, 1992.

[44] J. de Kleer and O. Raiman, "Trading off the costs of inference versus probing in diagnosis," in *Proc. IJCAI*, 1995, pp. 1736–1741.

[45] J. Kolodner, *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann, 1993.

[46] J. Kolodner and D. B. Leake, "A tutorial introduction to case-based reasoning," in *Case Based Reasoning: Experiences, Lessons, and Future Directions*, D. B. Leake, Ed. Menlo Park, CA: AAAI Press, 1996.

[47] G. W. Leng and L. K. Teen, "ESPCRM—An expert system for personal computer repair and maintenance," *Eng. Applicat. Artif. Intell.*, vol. 5, no. 2, pp. 121–133, 1992.

[48] J. H. Lim *et al.*, "INSIDE: A connectionist-based diagnostic expert system that learns incrementally," in *Proc. Int. Joint Conf. Neural Networks*, vol. 2, 1991, pp. 1693–1698.

[49] Z.-Q. Liu and F. Yan, "Case-based diagnostic system using fuzzy neural network," in *Proc. IEEE Conf. Neural Networks*, vol. 6, 1995, pp. 3107–3112.

[50] G. F. Luger and W. A. Stubblefield, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Reading, MA: Addison-Wesley, 1998.

[51] Y. Maidon *et al.*, "Diagnosis of multifaults in analogue circuits using multilayer perceptrons," *Proc. Inst. Elect. Eng., Circuits Devices Syst.*, vol. 144, no. 3, pp. 149–154, June 1997.

[52] L. P. Maguire *et al.*, "Intelligent test and repair of microprocessor-based products," in *Proc. Joint Conf. Inform. Syst.*, 1998, pp. 283–286.

[53] F. Mohammed, M. Marzouki, and M. H. Touati, "FLAMES: A fuzzy logic ATMS and model-based expert system for analog diagnosis," in *Proc. Euro. Design Test Conf.*, 1996, pp. 259–263.

[54] F. Mohammed and M. Marzouki, "Test and diagnosis of analog circuits: When fuzziness can lead to accuracy," *J. Electron. Testing*, vol. 9, pp. 203–216, 1996.

[55] I. Mozetic, "Hierarchical model-based diagnois," *Int. J. Man-Mach. Stud.*, vol. 35, no. 3, pp. 329–362.

[56] J. H. Murphy and B. J. Kagle, "Neural network recognition of electronic malfunctions," *J. Intell. Manufact.*, pp. 205–216, 1992.

[57] B. D. Netten and R. A. Vingerhoeds, "Automatic fault tree generation," *IFAC Intell. Emerging Contr. Technol.*, pp. 173–178, 1994.

[58] M. Nicolaidis and Y. Zorian, "On-line testing for VLSIA—Compendium of approaches," *J. Electron. Testing: Theory Applicat.*, vol. 12, pp. 7–20, 1998.

[59] Y. Peng and J. A. Reggia, "A probabilistic causal model for diagnostic problem solving—Part I: Integrating symbolic causal inference with numeric probabilistic inference," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 146–162, Mar./Apr. 1987.

[60] ——, "A probabilistic causal model for diagnostic problem solving—Part II: Diagnostic strategy," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 395–406, May/June 1987.

[61] C. Preist, "Deploying a model-based diagnostic system in the manufacturing test environment," in *Proc. IEE Colloq. Artif. Intell. Manufact.*, London, U.K., 1997, pp. 1–4.

[62] E. Rich and K. Knight, *Artificial Intelligence*. New York: McGraw-Hill, 1991.

[63] J. G. Rowland and L. C. Jain, "Knowledge based systems for instrumentation diagnosis, system configuration and circuit and system design," *Eng. Applicat. Artif. Intell.*, vol. 6, no. 5, pp. 437–446, 1993.

[64] T. Satyanarayana, G. Subramanyam, and K. V. Rama Rao, "Implementing an expert system for fault diagnosis of electronic equipment," *Eng. Applicat. Artif. Intell.*, vol. 8, no. 3, pp. 355–364, 1995.

[65] D. Sauter *et al.*, "Fault diagnosis in systems using fuzzy logic," in *Proc. First Euro. Congr. Fuzzy Intell. Technol.*, 1993, pp. 781–788.

[66] B.-H. Alice Seng and B.-H. Tay, "A practical expert system for color TV diagnostic and repair," in *Proc. SPICIS*, 1994, pp. 448–453.

[67] J. W. Sheppard, "Explanation-based learning with diagnostic models," in *Proc. IEEE AutoTestCon Conf.*, 1992, pp. 159–166.

[68] M. Shirley and R. Davis, "Generating distinguishing tests based on hierarchical models and symptom information," in *Proc. IEEE Int. Conf. Comput. Design: VLSI in Comput.*, 1983, pp. 455–458.

[69] Y. Shimomura *et al.*, "Development of self-maintenance photocopiers," *AI Mag.*, vol. 16, no. 4, pp. 41–53, Winter 1995.

[70] J. W. Sheppard, "Inducing diagnostic inference models from case data," in *Research Perspectives and Case Studies in System Test and Diagnosis*, J. W. Sheppard and W. R. Simpson, Eds. Boston, MA: Kluwer, 1998.

[71] W. R. Simpson and J. W. Sheppard, *System Test and Diagnosis*. Boston, MA: Kluwer, 1994.

[72] ——, "Managing conflict in system diagnosis," *IEEE Computer*, vol. 31, no. 3, pp. 69–76, Mar. 1998.

[73] H. Sugimatsu *et al.*, "A CBR application: Service productivity improvement by sharing experience," in *Proc. IEEE Symp. Emerging Technol. Factory Automat.*, 1994, pp. 132–141.

[74] K. A. E. Totton and P. R. Limb, "Experience in using neural networks for electronic diagnosis," in *Proc. 2nd Int. Conf. Artif. Neural Networks*, 1991, pp. 115–118.

[75] S. G. Tzafestas and P. J. Dalianis, “Fault diagnosis in complex systems using artificial neural networks,” in *Proc. IEEE 3rd Conf. Contr. Applicat.*, 1994, pp. 877–882.

[76] R. Uthurusamy *et al.*, “Extracting knowledge from diagnostic databases,” *IEEE Expert*, pp. 27–38, Dec. 1993.

[77] L. A. Zadeh, “Fuzzy sets,” *Inform. Contr.*, vol. 8, pp. 338–353, 1965.

[78] ——, “Fuzzy algorithms,” *Inform. Contr.*, vol. 12, pp. 94–102, 1968.

[79] ——, “Outline of a new approach to the analysis of complex systems and decision making approach,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 28–44, Jan. 1973.

[80] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, NJ: IEEE Press, 1990.

[81] W. R. Simpson and J. W. Sheppard, “Encapsulation and diagnosis with fault dictionaries,” in *Proc. IEEE AutoTestCon*, 1996, pp. 441–445.

[82] B. Chess and T. Larrabee, “Creating small fault dictionaries,” *IEEE Trans. Comput.-Aided Design Integrated Circuits Syst.*, vol. 18, pp. 346–356, Mar. 1999.

[83] D. Mittelstadt, R. Paasch, and B. D’Ambrosio, “Application of a Bayesian network to integrated circuit tester diagnosis,” *Proc. ASME*, vol. 1, pp. 391–403, 1994.

[84] A. Darwiche, “Model-based diagnosis under real-world constraints,” *AI Mag.*, vol. 21, no. 2, pp. 57–73, 2000.

[85] O. Dressler and P. Struss, “The consistency-based approach to automated diagnosis of devices,” in *Principles of Knowledge Representation*, G. Brewka, Ed. Stanford, CA: CSLI, 1996, pp. 267–311.

[86] W. R. Simpson and J. W. Sheppard, “System complexity and integrated diagnosis,” *IEEE Des. Test Comput.*, pp. 16–30, Sept. 1991.

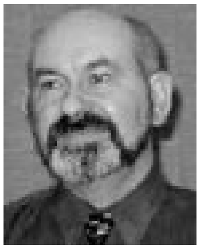
[87] D. Gartner and J. W. Simpson, “An experiment in encapsulation in system diagnosis,” in *Proc. AutoTestCon*, 1996, pp. 468–472.

[88] W. R. Simpson and J. W. Sheppard, “An intelligent approach to automatic test equipment,” in *Proc. Int. Test Conf.*, 1991, pp. 419–425.

[89] J. Taylor, “A standard for test and diagnosis,” in *Research Perspectives and Case Studies in System Test and Diagnosis*, J. W. Sheppard and W. R. Simpson, Eds. Boston, MA: Kluwer, 1998.

William G. Fenton (A’95) is currently pursuing the Ph.D. degree in the Intelligent Systems Engineering Laboratory, University of Ulster, Derry, U.K.

He is the Technical Director of International Test Technologies Ltd., Letterkenny, U.K., a leader in functional test and diagnostic solutions for microprocessor and DSP-based circuit boards. His research project relates to the application of intelligent techniques to the test and diagnosis of microprocessor and DSP boards and to the rapid deployment of artificial intelligence-based test and diagnosis techniques in the electronics manufacturing industry.



T. M. McGinnity (M’82) received the 1st Class Honors degree in physics from the New University of Ulster, Coleraine, U.K., and the Ph.D. degree from the University of Durham, Durham, U.K.

He is a Professor in the Intelligent Systems Engineering Laboratory and Head of the School of Computing and Intelligent Systems, Faculty of Informatics, University of Ulster, Derry, U.K. He has 22 years experience in teaching and research in electronic engineering. He founded and currently leads the research activities of the Intelligent Systems Engineering Laboratory at the University of Ulster. He is the author or co-author of over 80 research papers. His current research interests relate to the study and implementation of hybrid intelligent systems and in particular the creation of intelligent embedded systems with adaptation, reasoning, and evolving capabilities. Specific interests relate to intelligent techniques in reconfigurable computing systems, intelligent test and repair of embedded microprocessor systems, bioinspired evolvable hardware, and self-organizing intelligent systems.

Dr. McGinnity is a Fellow of the IEE and a Chartered Engineer.

Liam P. Maguire received the M.Eng. and Ph.D. degrees in electrical and electronic engineering from the Queen’s University of Belfast, Belfast, U.K., in 1988 and 1991, respectively.

He is a Lecturer in the School of Computing and Intelligent Systems, University of Ulster, Derry, U.K. He is also a Member of the Intelligent Systems Engineering Laboratory at the University of Ulster. His current research interests relate to the creation, design, and implementation of intelligent systems and, in particular, hybrid approaches embracing learning, reasoning, and evolving characteristics with applications in the embedded systems domain. He is the author or co-author of over 50 research papers and has contributed two book chapters in advanced texts.