AI Magazine Volume 16 Number 4 (1995) (© AAAI)

AGETS MBR

An Application of Model-Based Reasoning to Gas Turbine Diagnostics

Howard A. Winston, Robert T. Clark, and Gene Buchina

■ A common difficulty in diagnosing failures within Pratt & Whitney's F100-PW-100/200 gas turbine engine occurs when a fault in one part of a system—comprising an engine, an airframe, a test cell, and automated ground engine test set (AGETS) equipment—is manifested as an out-of-bound parameter elsewhere in the system. In such cases, the normal procedure is to run AGETS self-diagnostics on the abnormal parameter. However, because the self-diagnostics only test the specified local parameter, it will pass, leaving only the operators' experience and traditional fault-isolation manuals to locate the source of the problem in another part of the system. This article describes a diagnostic tool (that is, AGETS MBR), designed to overcome this problem by isolating failures using an overall system troubleshooting approach. AGETS MBR was developed jointly by personnel at Pratt & Whitney and United Technologies Research Center using an AI tool called the qualitative reasoning system (QRS).

A common difficulty in diagnosing failures within Pratt & Whitney's (PW) F100-PW-100/200 gas turbine engine occurs when a fault in one part of a system—comprising an engine, an airframe, a test cell, and automated ground engine test set (AGETS) equipment—is manifested as an out-of-bound parameter elsewhere in the system. In such cases, the normal procedure is to run AGETS self-diagnostics on the abnormal parameter. However, because the self-diagnostics only test the parameter specified, it will pass because parameter tests are local tests that cannot uncover malfunctions in other parts of the system. At this point, only the operators' experience and traditional fault-isolation manuals can be used to try to resolve the problem.

Task Description

In this section, we identify the task that is performed and the problem for which we required a solution, the objectives of the proposed application, and the reasoning behind using AI. We also discuss other solutions to the problems that were considered.

Task Performed and Problem Solved

AGETS is used by the United States Air Force and the Air National Guard to test, trim, and diagnose problems with Pratt & Whitney (PW) F100-PW-100 and F100-PW-200 jet engines. Sixty-six AGETS units exist worldwide and have been in operation since 1985. AGETS measures over 240 parameters from the engine, connected test equipment, and itself. These parameters include pressures, temperatures, rotational speeds, voltages, resistances, and discrete signals. Many of the monitored parameters originate from engine sensors, pass through AGETS for measurement, and continue on to the engine electronic control (EEC) or the unified fuel control (UFC). The presence of the EEC and the UFC complicates the diagnostic process because they can attempt to compensate for abnormal parameter deviations and, thus, mask fault symptoms. In addition, erroneous signals because of operator error or hardware malfunction can cause the control system to react to nonexistent problems. The number of parameters and the potential for control-system interference provide ample opportunity for difficult problems to arise.

The fault trees found in the F100 and AGETS paper maintenance manuals (Air Force 1988, 1984) are typical of the most widely used approach for aiding maintainers during troubleshooting. Fault trees present the maintainer with a sequence of troubleshooting steps designed to isolate a predefined set of failures. However, the fault tree approach suffers from a number of shortcomings. In particular, paper fault trees are unable to diagnose faults with novel or unforeseen symptoms, inflexible in the sequence of tests required to isolate a fault, cumbersome to use, and costly to update.

An alternative (not currently used by AGETS operators) to the paper-based fault tree is the rule-based expert system. Rule-based diagnostic expert systems use experts' knowledge about the relations between symptoms and causes encoded in the form of if-then rules. Sequences of troubleshooting steps are computed dynamically by performing logical inferences over these rules.

However, as problem complexity increases, development and maintenance of rule bases can become extremely difficult and costly. Moreover, because the rules in the knowledge base are derived from the past experience of experts, rule-based systems, like fault trees, are limited to diagnosing faults that have been experienced before and have well-established characteristics.

Objectives of the Application and Motivation for an AI Solution

To overcome the shortcomings of fault-based approaches to diagnostics, a new approach to diagnostics based on representing normal behavior was used (that is, *qualitative model-based reasoning*). AGETS MBR is a model-based reasoning system that uses qualitative models of the normal behavior of an engine, airframe, AGETS, and various pieces of test equipment to troubleshoot problems by isolating failures to one of these system components. The qualitative models employed by AGETS MBR were developed using a patented software system, qualitative reasoning system (QRS), developed at United Technologies Research Center (UTRC).

Other Solutions

Prior to the development of AGETS MBR, diagnostic procedures were developed separately and at different times for the engine, AGETS, test cell, airframe, and controller subsystems, but no procedure existed that could diagnose problems in the overall global system. In effect, each of the subsystem diagnostics was designed as if that subsystem existed independently of the others. Thus, no formalized or mechanized process initially determined which subsystem(s) could be malfunctioning and accounted for how subsystem interactions shaped fault manifestations.

The gap caused by the lack of a tie-in between independent subsystem diagnostics made it necessary to train and support a large number of engineers (both from PW and the Air Force) who could use their experience to solve global diagnostic problems. In addition, the resulting large personnel costs were aggravated by a high turnover rate that diluted the available experience base.

To address this problem, test procedures in the form of manual technical orders were proposed. However, it was realized that the complexity of subsystem interactions would make it difficult to assure consistency and completeness in a set of written technical orders for global diagnostics. Moreover, a flexible procedure was needed to handle different system configurations.

Thus, to reduce the costs to the Air Force of providing telephone support for AGETS global system problems and formalize and preserve the experience base derived from handling these problems, an automated model-based approach to diagnostics was desirable. The existence of QRS technology at UTRC and the intuitive match between its representations and the way in which engineers conceptualized the structure and operation of AGETS and other components provided the motivation to develop AGETS MBR.

Application Description

In this section, we discuss the AGETS MBR system; the QRS AI tool used to develop AGETS MBR, including an example; and the modeling methodology. We also outline some of the lessons gained during the design and building of AGETS MBR and the application's use and payoff.

AGETS MBR

AGETS MBR was developed to assist field engineers and technicians at Kelly Air Force Base in Texas to troubleshoot problems experienced by AGETS users while testing F100 engines. The major part of AGETS MBR is a set of qualitative models of the testing environment that includes AGETS, the F100 engine, and various pieces of test-cell equipment. The purpose of AGETS MBR is to troubleshoot problems encountered during steady-state engine testing using AGETS. Problems meeting the following criteria were specifically excluded from AGETS MBR requirements: (1) engine fuel system problems, (2) engine ignition system problems, (3) backup control problems, (4) engine start problems, (5) engine static problems (that is, problems in which the engine is not running because, for example, it cannot be started), (6) engine oil system problems, (7) subtle engine performance problems (for example, fuel consumption degradations as a result of component wear), (8) problems for which existing diagnostic procedures are capable of directly isolating causes given their associated symptoms, (9) technical order or AGETS hardware-software design problems whose solutions have been implemented, (10) past problems for which insufficient documentation exists to positively identify causes, (11) Air Force supply system problems (that is, part procurement–related difficulties), (12) problems for which AGETS self-diagnostics (or other troubleshooting procedures in technical orders) are capable of isolating causes, (13) software problems, and (14) detailed component-level engine troubleshooting.

The goal of diagnosis with AGETS MBR is to isolate failures to the F100 engine, AGETS, airframe, or test-cell subsystems. Although AGETS MBR can troubleshoot within these modules (their associated qualitative models are hierarchically constructed from detailed models of their constituents), existing technical manuals are used for component-level troubleshooting to avoid potential conflicts with the order of established test procedures.

Three model configurations were developed: (1) uninstalled (encompassing AGETS, F100 engine, and M-37/T-20 test-cell components), (2) installed (encompassing AGETS, engine, test cell, and F- 15 or F-16 airframe components), and (3) stand alone (encompassing AGETS, engine, and airframe components). The goal of troubleshooting is to isolate a failure to a major subsystem (that is, AGETS, engine, test cell, or airframe) because no diagnostic aids exist for the global system except for the knowledge of experienced troubleshooters. Furthermore, many problems encountered can seemingly be attributed to engine or test-cell components but in fact originate in AGETS. As noted earlier, more detailed troubleshooting relies on existing fault-isolation techniques because of a desire to minimize development cost and eliminate confusion in cases where AGETS MBR might conflict with established troubleshooting procedures.

Qualitative Reasoning System (QRS)

This section describes the AI tool, QRS, that was used to develop AGETS MBR. QRS (Clark, Gallo, and Hamilton 1994) is a software system, developed at UTRC, that is designed to support the development of diagnostic applications of qualitative reasoning. Qualitative reasoning, or more generally symbolic model-based reasoning, is a subfield of AI that is concerned with the computation of possible behaviors of a device from a qualitative model of its structure and function. A *qualitative model* is an abstract representation of a device that allows decisions to be made from a high-level understanding of a situation, without the need for specific quantitative details that might be either (1) unavailable, (2) misleading (because the device might be broken so that the precision of a quantitative model might be inappropriate), or (3) untimely to attain.

QRS is made up of two major components: (1) the qualitative model developer and (2) the qualitative reasoner. The *model developer* is a graphic user interface that helps application domain experts build and test qualitative models. The *qualitative reasoner* is responsible for determining behaviors of qualitative models and can perform many functions such as state generation, fault detection, diagnosis, troubleshooting, and fault tree generation. AGETS MBR, as delivered to the U.S. Air Force San Antonio Air Logistics Center (SA-ALC) at Kelly Air Force Base, consists of the qualitative reasoner performing the troubleshooting function using qualitative models of the F100-PW-100/200 engine, associated F-15 or F-16 airframe components, test-cell equipment, and AGETS.

To perform troubleshooting, QRS first uses constraint propagation on qualitative models to determine whether given symptoms correspond to a failure. Next, QRS uses hierarchical constraint suspension (Davis 1984) to determine which failures could have caused the current symptoms. For each member of this list of suspected component failures, QRS generates the predicted values for model parameters that have not yet been measured or observed.

To perform efficient troubleshooting, QRS uses a process called *intelligent test selection* to choose the next test or observation to request. Intelligent test selection uses knowledge of component failure rates, along with model predictions, to estimate probabilities of test outcomes and the extent to which each available test can be expected to isolate a failure. QRS chooses the test that has the greatest overall utility, considering such factors as the extent to which each test is expected to isolate a fault, the a priori probabilities of various component failures, and the cost of performing each test.

A Simple Example In this section, we describe a simple example to illustrate the

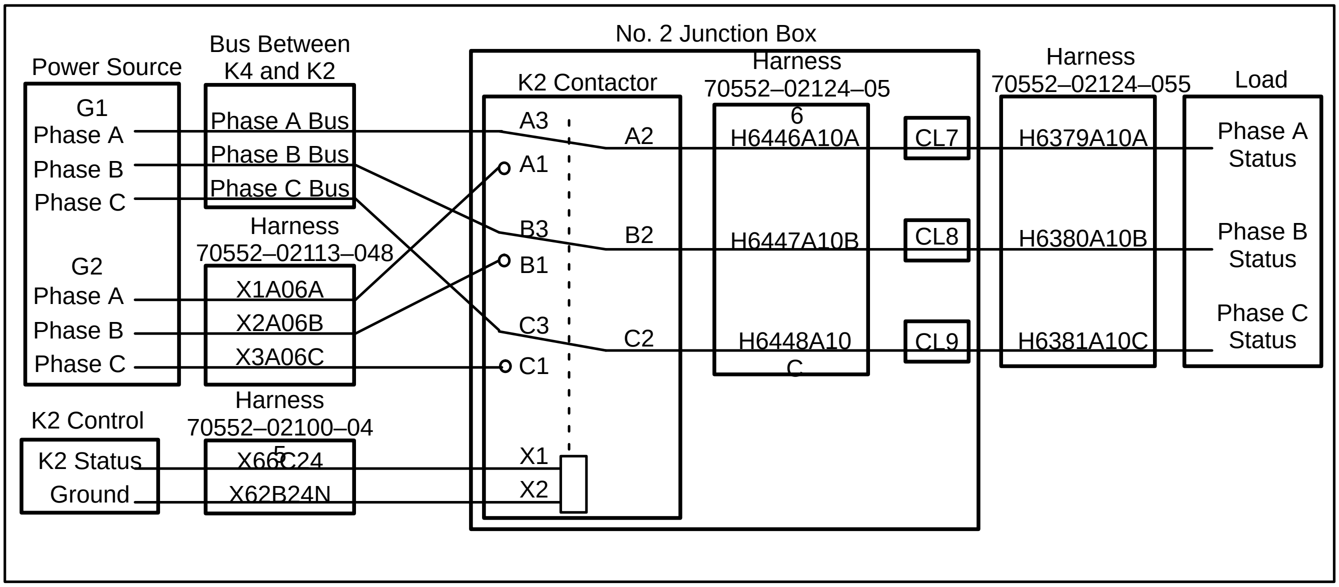


Figure 1. Schematic for CIRCUIT-1, Electric Power Distribution System.

troubleshooting process in QRS. The example is based on a portion of a generic electric power distribution subsystem. The example system, designated CIRCUIT-1, is depicted in figure 1. There are three input and three output variables of the system, as follows: (1) G1, shows generator #1 power status {on, off}; (2) G2, shows generator #2 power status {on, off}; (3) K2 control, controls whether the K2 contact selects G1 or G2 {G1, G2}; (4) phase A status, shows the status at phase A of the load {on, off}; (5) phase B status, shows the status at phase B of the load {on, off}; and (6) phase C status, shows the status at phase C of the load {on, off}.

A description of the normal behavior of the system is as follows: "If G1 is on and K2 control selects G1, then all phases should be on. If G2 is on and K2 control selects G2, then all phases should be on. Otherwise, all phases should be off." From the normal behavior models of CIRCUIT-1, the QRS state-generation algorithm can generate the possible normal behaviors of the system. These normal states are summarized in table 1.

Diagnosis Suppose that an operational checkout procedure is being performed on CIRCUIT-1. The first step in the checkout is to power on generator G1; power off generator G2; set K2 control to select G1; and observe phase A, B, and C status indicators. As seen in the third row of table 1, normal indications for this condition are phases A, B, and C on. Suppose, however, that phases A and B are on, and phase C is off. Using constraint propagation, QRS can determine that these symptoms are inconsistent with the normal behavior of the system.

QRS next uses hierarchical constraint suspension to determine which component failures can account for the symptoms. Assuming a single point of failure, QRS identifies the following components as suspects: load, wire H6381A10C, current limiter CL9, wire H6448A10C, K2 contact, phase C bus, or generator G1.

To better understand how QRS determines these suspects, consider the two components: current limiter CL9 (a suspect) and current limiter CL7 (not a suspect). To test current limiter CL9, its constraints are temporarily removed (suspended) from the constraint network. QRS then determines if a legal state can be generated with this new constraint network and the original failure symptoms. Suspending the constraints of CL9 removes the constraint between the input and output values (that is, voltages and currents) of CL9. Thus, the symptom that the phase C status indicator is off is consistent with the constraint network corresponding to a CL9 failure, and current limiter CL9 is identified as a suspect.

The constraint-suspension process is also used to test the current limiter CL7 failure hypothesis. After suspending the constraints associated with CL7, QRS attempts to generate a legal state consistent with the symptoms. This time, however, the conflict between the symptoms and the constraint network remains (that is, a CL7 failure cannot account for phase C status being off). Because a legal

Generator G1	Generator G2	K2 Control	Phase A Status	Phase B Status	Phase C Status
On	On	G1	On	On	On
On	On	G2	On	On	On
On	Off	G1	On	On	On
On	Off	G2	Off	Off	Off
Off	On	G1	Off	Off	Off
Off	On	G2	On	On	On
Off	Off	G1	Off	Off	Off
Off	Off	G2	Off	Off	Off

Table 1. Normal Value Assignments for CIRCUIT-1.

state cannot be generated for the CL7 failure hypothesis, CL7 is not identified as a suspect.

Fault Isolation As mentioned previously, to isolate the failure to a single component (or a minimal set of components), QRS uses a process called *intelligent test selection* that enables QRS to compute the utility of each applicable test. A test's *utility* is defined as its diagnostic power (that is, degree of fault isolation) divided by its associated cost.

For example, assume there are two tests that can be performed on CIRCUIT-1. Test 1 measures the voltage at terminal C3 of the K2 contactor. Test 2 measures the voltage between wire H6381A10C and the load. Given the initial list of suspects, test 1 partitions the list as follows: If the voltage is normal, then generator G1, phase C bus, and K2 contactor are eliminated. If the voltage is zero, then wire H6448A10C, current limiter CL9, wire H6381A10C, and the load are eliminated. Test 2 partitions the list of suspects as follows: If the voltage is normal, all components except the load are eliminated. If the voltage is zero, the load is eliminated.

To determine which test to select, the utility of each test must be computed. With the assumption of equal test costs and equal a priori component-failure probabilities, test 1 would have a higher utility than test 2. Intuitively, this higher utility is because test 1 partitions the remaining hypotheses into two sets of approximately equal size. Therefore, test 1 will remove about half the uncertainty of the diagnosis regardless of the outcome of the measurement. In contrast, because test 2 partitions the remaining hypotheses into two sets of unequal size, test 2 is most likely to remove just one-seventh of the uncertainty of the diagnosis.

In practice, the costs of different tests and the failure probabilities of various components can vary widely. Thus, the computed utility of test 2 might actually be higher than that of test 1. For example, if we assume that test 2 is very inexpensive compared to test 1, then test 2 might be selected before test 1, even though test 1 has greater diagnostic power. Alternatively, if the failure probability of the load is large relative to the failure probabilities of the other six suspects, the diagnostic power of test 2 might be greater than that of test 1. The process of computing the test utilities and selecting the most cost-effective test is repeated until only one hypothesis (or minimal set of hypotheses) remains.

Modeling Methodology

AGETS, F100 engine, airframe, and test equipment were modeled with qualitative variables, value spaces, and confluences. Qualitative variables differ from quantitative variables in that numeric values are not required. Instead qualitative variables typically have values drawn from value spaces such as {positive, zero, negative} or {high, low, normal}. For properties requiring a greater level of detail, additional landmarks can be inserted in the value space between zero and infinity. Landmarks can be used to represent numeric limits, values that bound regions of qualitatively distinct behavior, or simply on and off parameter settings. Qualitative variables are then combined into qualitative equations, or *confluences*, to describe normal device behavior. Confluences look very much like normal numeric or algebraic equations, except that parameters of unlike materials (for example, fuel and air) and unlike properties (for example, pressure and temperature) can be related directly with qualitative operators. (See de Kleer [1993], Forbus [1993], Kuipers [1993], de Kleer and Williams [1987],

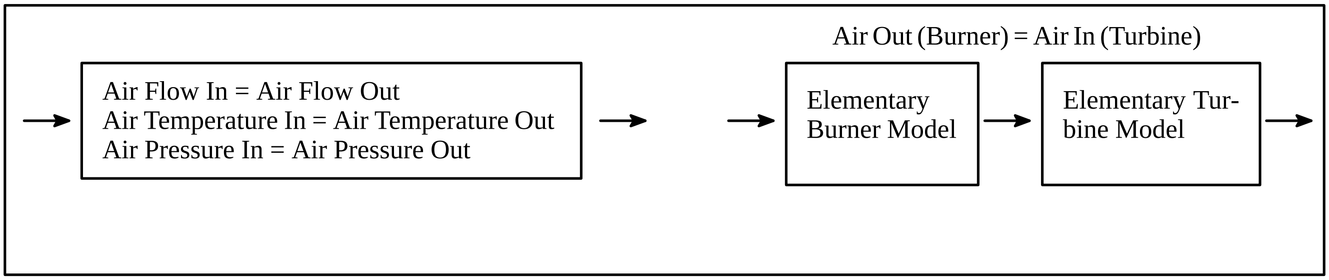


Figure 2. Elementary and Compound Models.

A. Elementary Burner Model. B. Compound Hot-Section Model.

Kuipers [1986], de Kleer and Brown [1984], Forbus [1984], Kuipers [1984] for more information about qualitative modeling, representation, and reasoning.)

The objective of modeling AGETS and other system components was to construct a functional representation, not necessarily a physical one. That is, logically related components were sometimes grouped together, even though they might exist in physically different parts of the overall system. For this purpose, QRS supports the representation of primitive components (without substructure) in the form of elementary qualitative models and complex components (having an internal structure) in the form of compound qualitative models. Functional groupings of related components were represented as compound models in AGETS MBR.

Three steps occur in the elementary model-building process: (1) identify relevant parameters, (2) determine which parameters are also input and output terminals, and (3) constrain the parameters with confluences. As an example, for a burner model, the relevant parameters are airflow in, air pressure in, air temperature in, fuel flow in, gas flow out, gas pressure out, and gas temperature out. All these parameters are terminals. The confluences are as shown in figure 2a.

Compound model construction simply involves connecting terminals of like material and property between elementary or compound models, as shown in figure 2b. (For brevity, air in or air out is an abbreviation for the parameters of airflow, temperature, and pressure in or pressure out, respectively.) This connection of input to output terminals continues with elementary and compound models until the top-level application model is completed. Unlike the ability to directly relate or equate parameters of dissimilar materials in confluences, only terminals composed of the same material can be connected because they essentially establish identity relations between parameters in different models.

Lessons Learned

From our experience designing and building AGETS MBR, we gained several insights into the application of AI technology:

First, the acceptance of AGETS MBR was enhanced by an important step taken at the start of the program. To ensure that the features and operation of the delivered system would meet the customers' expectations, we conducted a two-day joint-application design (JAD) session (Wood and Silver 1989) at Kelly Air Force Base to understand what the Air Force needed and how they wanted it delivered and to precisely define the application's capabilities. A JAD is a brainstorming process for eliciting a set of project requirements and specifications from a customer with only minimal vendor interference. (UTRC technologists were only permitted to answer technical feasibility questions.) This design session helped to preclude proposing solutions that modified the "nail to fit the hammer."

Second, the project was divided into two phases that minimized risk to the customer. The first phase developed a prototype system for a small AGETS subset that was demonstrated to the customer and checked against requirements that arose from the initial JAD session. New requirements also arose from interaction with the prototype, which ensured that the complete AGETS MBR diagnostic system delivered at the end of phase 2 met or exceeded Air Force expectations.

Third, AGETS MBR was developed on a Sun platform and delivered on a Pentium-based PC. Although the customer assigned no resource limitations to the required PC delivery platform, the performance of the delivered system was, nevertheless, slower than

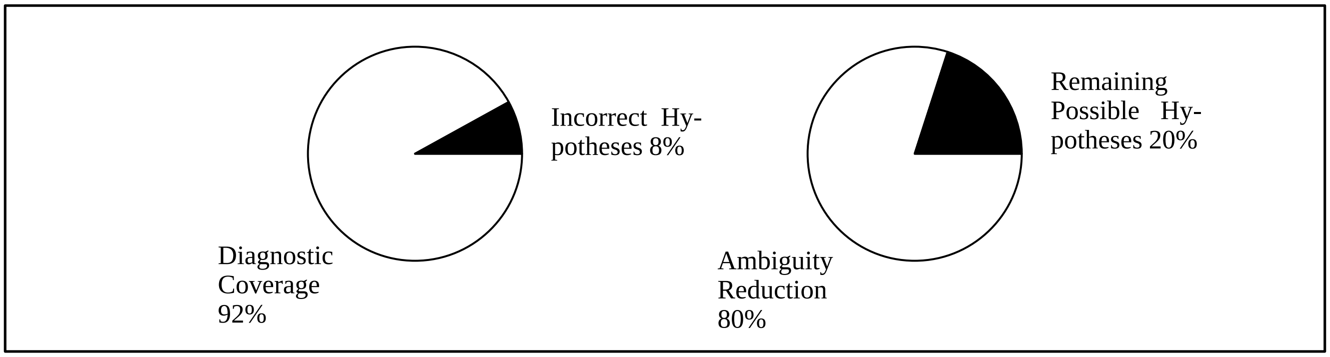


Figure 3. Diagnostic Coverage and Ambiguity Reduction.

the Air Force expected; thus, the system probably lost some acceptance. We have learned that the performance of an AI application is one of its most important features.

Application Use and Payoff

AGETS MBR has been in use by members of the SA-ALC and PW employees since June 1994. We tested AGETS MBR on the set of all telephone hot-line calls received at Kelly Air Force Base from worldwide AGETS field locations from 1 June to 12 October 1994. Thirty-one calls were received. Of these calls, 13 were acceptable for AGETS MBR troubleshooting as outlined in the subsection entitled AGETS MBR. In all 13 of these cases, given the initial symptoms, AGETS MBR detected discrepancies between the expected behavior and the actual behavior of the system under test. Additionally, each initial fault hypothesis list generated by AGETS MBR included the actual cause of the problem.

Because of the conditions outlined in the AGETS MBR subsection, the installed version of AGETS MBR is not used to isolate the failure beyond the highest-level modules in the system under test. To understand the true power and capabilities of the qualitative models and qualitative reasoner used by AGETS MBR and to evaluate the effectiveness of AGETS MBR (that is, what the system could do if not for the limited scope of how it is routinely used), the 13 cases identified previously were put through a more rigorous analysis. For these tests, AGETS MBR was permitted to diagnose down to the line replaceable unit (LRU) level of the system under test. The results of these tests showed whether AGETS MBR could localize the actual failure to one of its elementary models as well as what the final level of ambiguity reduction was from the fault hypothesis list generated from the given initial symptoms.

Table 2 summarizes the results of these detailed tests. *Diagnostic coverage* is defined as whether or not the actual cause was listed as a possible fault identified by AGETS MBR. In other words, the diagnostic coverage is positive if AGETS MBR could identify the actual failure; otherwise, it is negative. AGETS MBR was able to identify the failure in 12 of the 13 cases, leading to an overall diagnostic coverage of 92 percent. The other important statistic revealed in the table is that the average ambiguity reduction is 80 percent. *Ambiguity reduction* is defined as the percentage of components that are removed from consideration because they cannot cause the fault given the symptoms as defined in the case. These statistics are displayed in figure 3.

As of 15 April 1995, seventy-seven telephone calls were received, of which 23 were within the design scope of AGETS MBR. Faults were detected in all 23 cases, and the preliminary hypothesis list included the actual cause in all these cases.

Because of the low frequency of trouble calls and the decision to limit diagnosis capability to high-level system modules, the benefits of AGETS MBR are hard to quantify. As such, efforts are under way to relax the operational requirements of AGETS MBR to diagnose to the LRU level. Postdelivery experience has shown that many difficult problems occur within AGETS. Allowing AGETS MBR to troubleshoot such problems would make payoff calculations more tractable in terms of financial and labor metrics.

An unexpected benefit of system delivery was the decision to base the PW development engineer at Kelly Air Force Base as an on-site representative. The experience of working with QRS and AGETS MBR enhanced his troubleshooting skills by forcing him to look at familiar situations in unfamiliar ways. Many times during development, experts learned something new about troubleshooting AGETS and F100 engines. PW Government Engine and Space Propulsion (PW-GESP) also gained

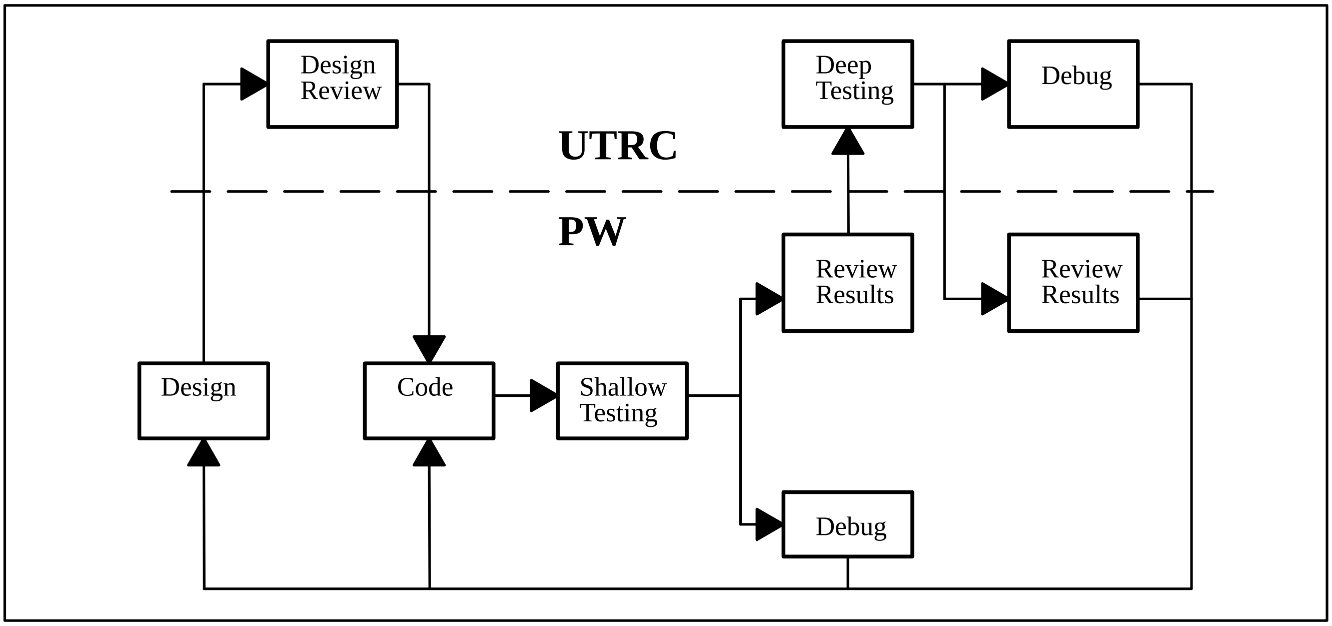


Figure 4. UTRC-PW Software Development Process.

Case	Diagnostic Coverage	Number of Hypotheses	Possible Causes	Ambiguity Reduction (%)
Eglin	Yes	7	259	97%
P064006	Yes	5	259	98
P074002	Yes	18	259	93
P074004n	Yes	101	259	61
P074004p	Yes	87	259	66
P074005	Yes	74	259	71
P084003	Yes	97	259	63
P084007	Yes	20	259	92
P084007t	Yes	94	259	64
P084009	Yes	24	259	91
P084014	Yes	42	259	84
P094003	Yes	42	259	84
P074003	No	N/A	264	N/A
Average	92%	51	259	80

Table 2. Detailed Results.

experience with AI diagnostic systems. Specifically, an engineer acquired expertise in modeling complex systems for making accurate diagnoses. PW-GESP plans to use this expertise in future projects.

Application Development and Deployment

In this section, we discuss the basic model-development process, the deployment of AGETS MBR, and system maintenance.

Development Processes

The basic model-development process iterated testing and redesign processes between UTRC and PW. Figure 4 shows how model-development steps were coordinated between these two organizations. PW provided initial model designs based on its domain knowledge that were reviewed by UTRC to ensure consistency with other models and compatibility with QRS modeling primitives. PW then coded the models into QRS and tested their fault-detection capability (that is, shallow

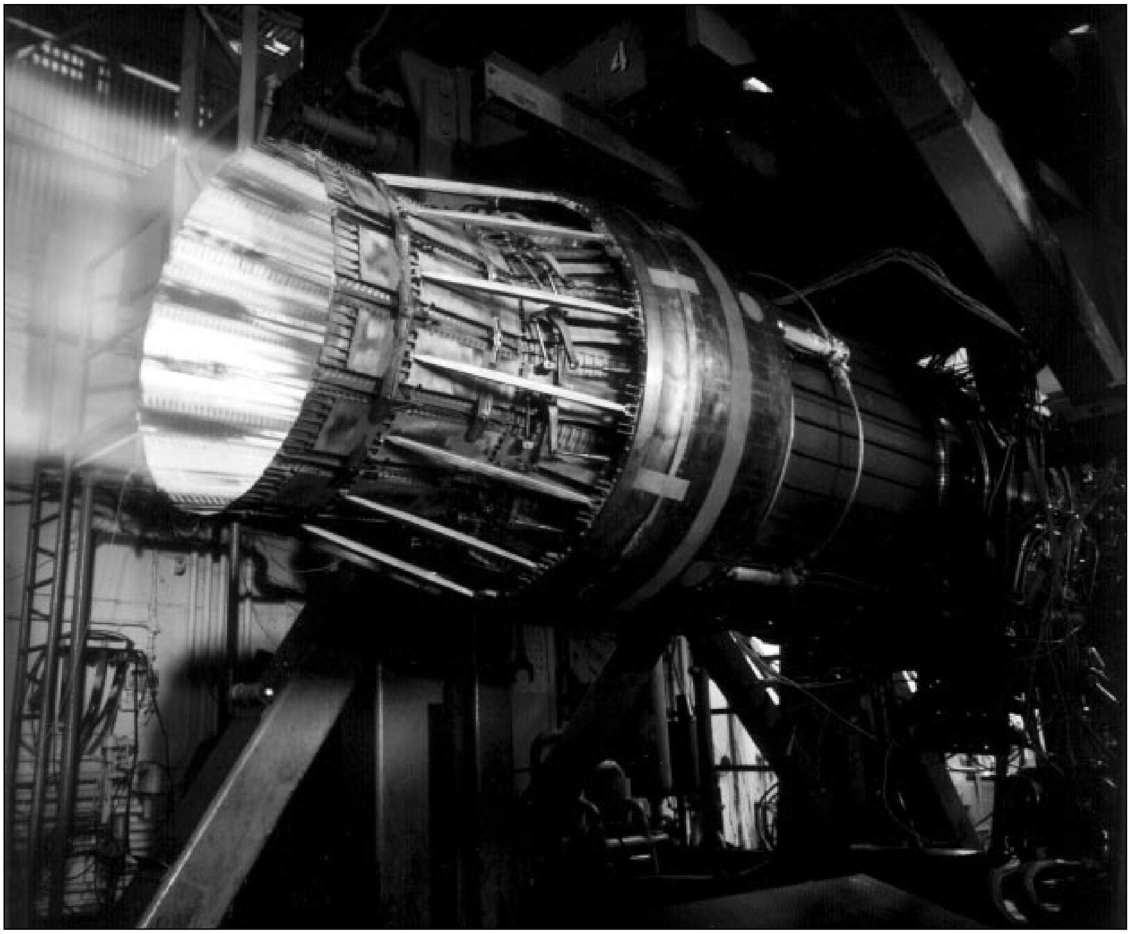


Figure 5. Pratt & Whitney's F100-PW-220 Turbofan Engine Undergoes Sea-Level Testing.

testing). Fault detection only determined whether or not faults existed based on canonical sets of corresponding symptoms.

If a review by PW of shallow testing results was satisfactory, UTRC proceeded with detailed hypothesis testing (that is, deep testing). Hypothesis testing determined whether diagnostic procedures applied to the models could find correct hypothesis sets. Based on PW's review of deep testing results, further model debugging was done at UTRC. PW then proceeded to redesign and recode the new models. If a review by PW of shallow testing results was unsatisfactory, PW would debug, redesign, and recode the models. This procedure iterated until PW determined that the results of deep testing were satisfactory.

The AGETS MBR models were developed over the course of approximately 18 months and at a cost of 15 person-months and two developers (one from UTRC and one from PW) at any given time. AGETS MBR encompasses three AGETS configurations (unique models are specific to a particular configuration): (1) installed, comprising 1398 elementary and 242 compound models (212 models are unique); (2) uninstalled, comprising 1368 elementary and 229 compound models (181 models are unique); and (3) stand alone, comprising 1362 elementary and 235 compound models (209 models are unique).

An existing model of a commercial turbofan engine (Winston et al. 1991) was used as the starting point for the gas-path portion of the F100 engine model (figure 5). The confluences were derived from the basic thermodynamic behavior of gas-turbine components such as compressors, burners, and turbines. The electric and hydromechanical control of the F100 proved to be especially challenging to model. Many iterations were required to eliminate static instability in the control-system model. The AGETS models were designed from electric schematics in the AGETS maintenance manual (Air Force 1988). Although AGETS contains numerous components, from wires to computer cards, similarity permitted a high degree of model reuse. Additionally, software routines were used to automate construction of large compound models. A small number of aircraft and test-cell component models were developed at relatively small expense.

After a model was constructed, it was tested for accuracy using the state-generation and diagnosis utilities in QRS. First, the number of legal states generated for a set of input conditions was determined. In most cases, exactly one generated state was desirable. In certain cases, more than one legal state was needed to accurately describe the function of a component. After successful completion of state-generation testing, typical system problems were simulated, first to ensure fault detection (that is, shallow testing), then to compile hypothesis lists (that is, deep testing). These hypothesis lists were checked against historical evidence and reviewed by domain experts. As explained earlier, deviations from expected results usually caused an iteration in the model design.

Test procedure design and implementation were accomplished concurrently with model testing. The majority of test procedures used by AGETS MBR were of two main types: (1) parameter observations and (2) AGETS hardware checks. A small number of test procedures involved engine observations. As before, a desire to avoid user confusion by conflicting with established troubleshooting methods was the reason only a few engine-related test procedures were used.

The final process in the AGETS MBR development was converting the entire system from a UNIX platform, on which it was developed, to an IBM-compatible PC platform. AGETS MBR was delivered on an IBM-compatible PC using a 60-megahertz Pentium processor and 64 megabytes of random-access memory. The PC platform was chosen at the customer's request for commonality with existing hardware.

Deployment Process

The AGETS MBR system was intended to be used easily by personnel at the SA-ALC at Kelly Air Force Base. It was designed to be user friendly by incorporating an easy-to-use graphic user interface along with a full-featured online help system. Because of these objectives, the deployment process was similar to that of a commercial software package.

AGETS MBR was successfully installed at the SA-ALC in June 1994. Installation consisted of procuring the hardware, obtaining needed software licenses, and loading the completed QRS models. Following installation, a two-week acceptance test and training period was completed to the customer's satisfaction. Sample and actual problems, as well as customer test cases, were used to qualify the system during the acceptance test. Training consisted of supplying user manuals and conducting tutorial sessions. After completion of training, a PW employee who could answer questions about AGETS MBR remained on site at SA-ALC.

Maintenance

As of this writing, there have been two updates to the AGETS MBR software. The first update involved miscellaneous user interface improvements, and the second update involved a major performance improvement. Neither update was planned as maintenance, and new releases were not included in the customer's original contract. As such, new updates are not planned at this time. However, several improvements to the AGETS MBR software have been identified by the customer. These enhancements include extending coverage to previously excluded systems (for example, engine fuel system, oil system), including decision trees automatically compiled from AGETS MBR models for more efficient performance, providing an extended tutorial system, and deploying the AGETS MBR software to remote field sites.

PW complex equipment support engineers are responsible for maintaining AGETS MBR qualitative models—the knowledge base that supports diagnosis of the actual system. Because qualitative models are based on normal device behavior, which does not often change over time, it is not expected that many changes to the knowledge base will be required. This expectation is in contrast to fault-based approaches (for example, fault trees or shallow rule-based systems) where new failure modes can frequently be discovered. However, when changes to the knowledge base are needed because of incorrect models of device behavior or changes to the actual system, updating the application should be fairly straightforward in that one qualitative model can easily be substituted for another qualitative model without disrupting the remaining models in the system.

Conclusion

In many cases, problems experienced during F100 engine testing with AGETS are difficult to troubleshoot for any or all of the following reasons: (1) the volume and complexity of AGETS measurements, (2) possible interference by the engine's electrical and hydromechanical control systems, and (3) the lack of a formal troubleshooting aid for the testing system.

AGETS MBR, using QRS software, provides support personnel at Kelly Air Force Base with a diagnostic aid that encompasses the entire engine-airframe and testing system. In addition, AGETS MBR does not suffer from the many pitfalls of traditional fault trees. These considerations and the results of AGETS MBR testing motivate the shift from fault tree–based approaches to normal behavior approaches, as used in AGETS MBR.

Acknowledgments

The authors would like to thank the following people who contributed to the success of the AGETS MBR system. Tom Hamilton has led UTRC's QRS technology development for over 10 years and helped to establish and define the AGETS MBR program. At Pratt & Whitney (PW), Eric Meyer was a lead engineer on the AGETS program who initiated the AGETS MBR collaboration with UTRC and contributed to its early development, Scott Connally provided valuable information about AGETS operation, and Bob Wilkoff developed a complementary diagnostic case-based system for handling variations of known failure modes. Don Wade (PW) moderated the early joint-application design session at Kelly Air Force Base.

References

Air Force. 1988. Organizational and Intermediate Maintenance Automated Ground Engine Test Set PWA52105, Technical Order 33D4-6-690-2, 6-1–6-266. Washington, D.C.: Government Printing Office.

Air Force. 1984. Intermediate Troubleshooting and Engine Subsystem Description, Aircraft Engine. USAF Model F100-PW-100(3), Technical Order 2J-F100-21-3, WP 052, 8–121. Washington, D.C.: Government Printing Office.

Clark, R. T.; Gallo, S.; and Hamilton, T. P. 1994. Qualitative Reasoning: Theory and Applications. In Proceedings of the Thirty-Second AIAA Aerospace Sciences Meeting, AIAA-94-0296. New York: American Institute of Aeronautics and Astronautics.

Davis, R. 1984. Diagnostic Reasoning Based on Structure and Behavior. *Artificial Intelligence* 24:347–410.

de Kleer, J. 1993. A View on Qualitative Physics. *Artificial Intelligence* 59:105–114.

de Kleer, J., and Brown, J. S. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence* 24:7–83.

de Kleer, J., and Williams, B. C. 1987. Diagnosing Multiple Faults. *Artificial Intelligence 32*(1): 97–130.

Forbus, K. 1993. Qualitative Process Theory: Twelve Years After. *Artificial Intelligence* 59:115–123.

Forbus, K. 1984. Qualitative Process Theory. *Artificial Intelligence* 24:85–168.

Hamilton, T. P. 1988. HELIX: An Application of Qualitative Physics to Diagnostics in Advanced Helicopters. *International Journal for AI in Engineering* 3(3): 141–150.

Kuipers, B. J. 1993. Qualitative Simulation: Then and Now. *Artificial Intelligence* 59:133–140.

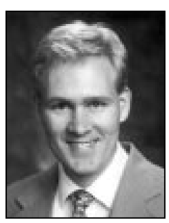
Kuipers, B. J. 1986. Qualitative Simulation. *Artificial Intelligence* 29(3): 289–338.

Kuipers, B. J. 1984. Commonsense Reasoning about Causality: Deriving Behavior from Structure. *Artificial Intelligence* 24:169–203.

Winston, H.; Sirag, D.; Hamilton, T.; Smith, H.; Simmons, D.; and Ma, P. 1991. Integrating Numeric and Symbolic Processing for Gaspath Maintenance. In Proceedings of the Twenty-Ninth AIAA Aerospace Sciences Meeting, AIAA-91-0501. New York: American Institute of Aeronautics and Astronautics.

Wood, J., and Silver, D. 1989. *Joint Application Design*. New York: Wiley.

Howard Winston received a bachelor's degree in physics at Rensselaer Polytechnic Institute in 1978. He received a master's degree in physics in 1980 and a Ph.D. in physics in 1984, both from Brown University. Winston then joined the Systems Science Department at Schlumberger-Doll Research in Ridgefield, Connecticut, where he participated in several AI research and development projects. In 1989, he joined United Technologies Research Center in East Hartford, Connecticut, where he has primarily been engaged in developing and applying machine learning, fuzzy logic, and qualitative physics technology.



Rob Clark is a system support engineer for the SunService Division of Sun Microsystems. He has been working on site at Pratt & Whitney since March 1995. His duties include system administration, programming, and problem resolution. Clark has also held various programming positions as well as taught undergraduate computer science courses at the University of Connecticut. Clark holds a master of science, as well as engineering degrees, from the University of Connecticut.

Gene Buchina is a field support engineer for Pratt & Whitney (PW) stationed at Kelly Air Force Base in San Antonio, Texas. Since joining PW in 1989, Buchina has been involved in jet engine simulation, experimental data analysis, test equipment development, and customer support. Buchina holds a bachelor of science in aerospace engineering from Pennsylvania State University.