AGETS MBR: An Application of Model–Based Reasoning to Gas Turbine Diagnostics

Howard A. Winston and Robert T. Clark[1]

United Technologies Research Center
411 Silver Lane
East Hartford, Connecticut 06108, USA
haw@utrc.utc.com

Gene Buchina

United Technologies – Pratt & Whitney Government Engine and Space Propulsion
P.O. Box 109600
West Palm Beach, Florida 33410–9600, USA
buchina@pwfl.com

Abstract

Currently, 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, airframe, test cell, and Automated Ground Engine Test Set (AGETS) equipment, is manifested as an out–of–bounds parameter elsewhere in that 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 paper describes a diagnostic tool (*i.e.*, 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 (PW) and United Technologies Research Center (UTRC) using an Artificial Intelligence tool called Qualitative Reasoning System (QRS).

Task Description

Task Performed and Problem Solved

Currently, 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, airframe, test cell, and Automated Ground Engine Test Set (AGETS) equipment, is manifested as an out–of–bounds parameter elsewhere in that 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.

AGETS is used by the U.S. Air Force and Air National Guard to test, trim and diagnose problems with Pratt & Whitney 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 Unified Fuel Control (UFC). The presence of the EEC and UFC complicate the diagnostic process because they can attempt to compensate for abnormal parameter deviations and thus mask fault symptoms. In addition, erroneous signals due to operator error or hardware malfunction can cause the control system to react to nonexistant 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 (Intermediate Troubleshooting ... 1984; Organizational and Intermediate ... 1988) are typical of the most widely used approach for aiding maintainers during troubleshooting. Fault trees present the maintainer

---

[1] Currently with Sun Microsystems–SunService Division
175 Capital Boulevard
Rocky Hill, Connecticut 06067, USA
Rob.Clark@East.Sun.COM

with a sequence of troubleshooting steps designed to isolate a pre-defined 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, since 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 (*i.e.*, qualitative model-based reasoning). AGETS MBR is such 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 had been 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 were designed as if that subsystem existed independently of the others. Thus, there was no formalized or mechanized process to initially determine which subsystem(s) could be malfunctioning and to account for how subsystem interactions shaped fault manifestations.

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

To address this problem, test procedures in the form of manual technical orders (TOs) 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 TOs for global diagnostics. Moreover, a flexible procedure was needed to handle different system configurations.

Thus, in order to reduce the costs to the Air Force of providing telephone support for AGETS global system problems and as a way to 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

AGETS MBR

AGETS MBR was developed to assist field engineers and technicians at Kelly Air Force Base (AFB) 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:

- Engine fuel system problems
- Engine ignition system problems
- Back-Up Control problems
- Engine start problems
- Engine static problems (*i.e.*, problems in which the engine is not running because, for example, it cannot be started)
- Engine oil system problems
- Subtle engine performance problems (*e.g.*, fuel consumption degradations due to component wear)
- Problems for which existing diagnostic procedures are capable of directly isolating causes, given their associated symptoms
- Technical order or AGETS hardware/software design problems whose solutions have been implemented
- Past problems for which insufficient documentation exists to positively identify causes

- Air Force supply system problems (*i.e.*, part procurement–related difficulties)
- Problems for which AGETS self–diagnostics (or other troubleshooting procedures in technical orders) are capable of isolating causes
- Software problems
- 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: *(i)* uninstalled (encompassing AGETS, F100 engine, and M–37/T–20 test cell components), *(ii)* installed (encompassing AGETS, engine, test cell, and F–15 or F–16 airframe components), and *(iii)* standalone (encompassing AGETS, engine, and airframe components). The goal of troubleshooting is to isolate a failure to a major subsystem (*i.e.*, AGETS, engine, test cell, or airframe) since 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 above, more detailed troubleshooting relies on existing fault isolation techniques due to 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 artificial intelligence tool, QRS, used to develop AGETS MBR. QRS (Clark, Gallo, & Hamilton 1994) is a software system, developed at UTRC, designed to support the development of diagnostic applications of qualitative reasoning. Qualitative reasoning, or more generally symbolic model–based reasoning, is a subfield of artificial intelligence 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 may be either *(i)* unavailable, *(ii)* misleading (because the device might be broken so that the precision of a quantitative model might be inappropriate), or *(iii)* untimely to attain.

QRS is made up of two major components: the qualitative model developer and the qualitative reasoner. The model developer is a graphical user interface which 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 at Kelly AFB, 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.

In order 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.

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.

A Simple Example. In this section, we describe a simple example to illustrate the troubleshooting process in QRS. The example is based on a portion of a generic electrical power distribution subsystem. The example system, designated CIRCUIT–1, is depicted in Figure 1. There are three inputs and three outputs to the system, as follows:

- *G1*, generator # 1 power status {On, Off}
- *G2*, generator # 2 power status {On, Off}
- *K2 Control*, controls whether the K2 contactor selects G1 or G2 {G1, G2}
- *Phase A Status*, shows the status at Phase A of the load {On, Off}

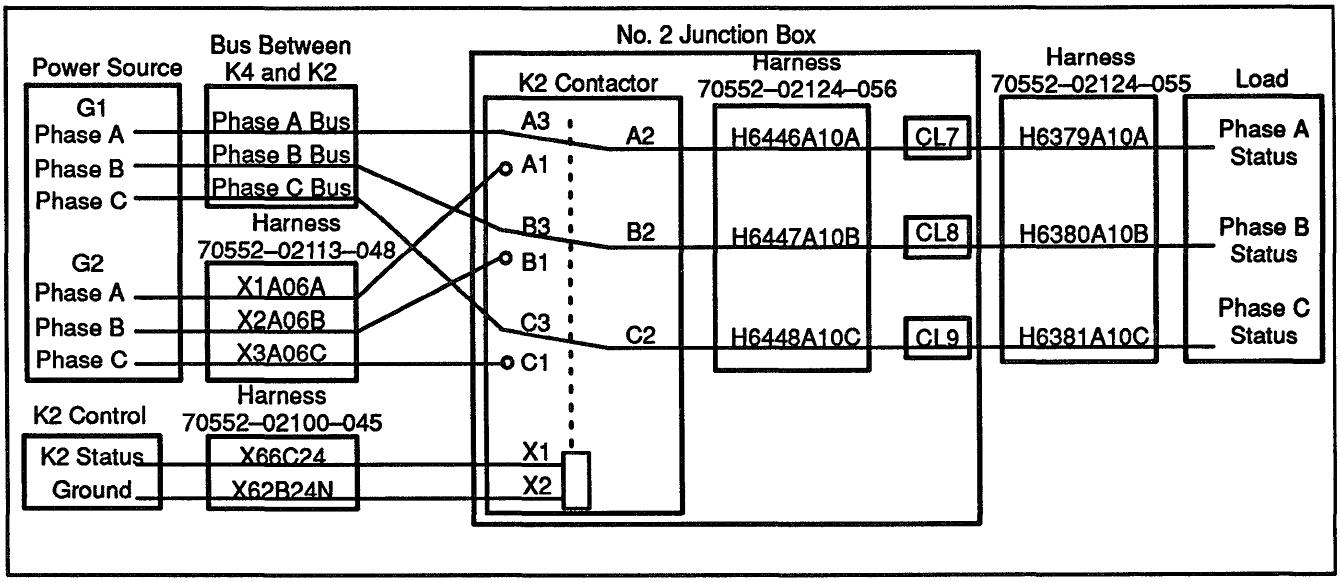


Figure 1. Schematic for CIRCUIT–1, an electrical power distribution subsystem.

- *Phase B Status*, shows the status at Phase B of the load {On, Off}
- *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 the 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 Contactor, 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 (*i.e.*, 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 (*i.e.*, a CL7 failure cannot account for Phase C status being Off). Since a legal 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 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 (*i.e.*, degree of fault isolation) divided by its associated cost.

For example, assume there are two tests that may 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 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. Assuming equal test costs and equal a priori component failure probabilities, TEST–1 would have a higher utility than TEST–2. Intuitively, this is because TEST–1 partitions the remaining hypotheses into two sets of approximately equal size. Therefore, TEST–1 will remove about half of the uncertainty of the diagnosis, regardless of the outcome of the measurement. In contrast, since 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 may vary widely. Thus, the computed utility of TEST–2 may 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 may 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 may 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 modelled with qualitative variables, value spaces, and confluences. Qualitative variables differ from quantitative variables in that numerical values are not required. Instead qualitative variables normally 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 numerical 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 numerical or algebraic equations, except that parameters of unlike materials (*e.g*,. fuel and air) and unlike properties (*e.g*., pressure and temperature) can be directly related with qualitative operators. (See (de Kleer & Brown 1984; de Kleer & Williams 1987; de Kleer 1993; Forbus 1984; Forbus 1993; Kuipers 1984; Kuipers 1986; Kuipers 1993) 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.

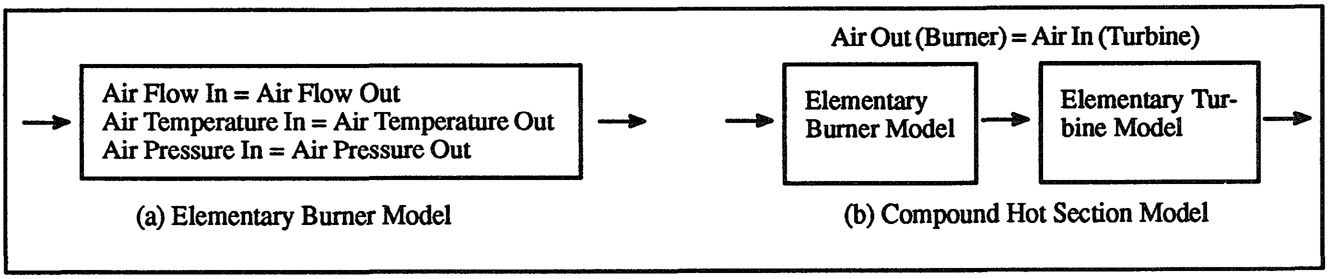
Three steps occur in the elementary model building process: *(i)* identify relevant parameters, *(ii)* determine which parameters are also input and output terminals, and then *(iii)* constrain the parameters with confluences. As an example, for a burner model, the relevant parameters are Air Flow In, Air Pressure In, Air Temperature In, Fuel Flow In, Gas Flow Out, Gas Pressure Out, and Gas Temperature Out. All of these parameters are terminals. The confluences are as shown in Figure 2(a).

Compound model construction simply involves connecting terminals of like material and property between elementary or compound models, as shown in Figure 2(b). (For brevity, Air In or Out is an abbreviation for the three Air Flow, Temperature, and Pressure In or Out parameters.) 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 dissimiliar materials in confluences, only terminals comprised 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, several insights were gained about the application of AI technology.

1) The acceptance of AGETS MBR was enhanced by an important step taken at the start of the program. In order to insure that the features and operation of the delivered system would meet the customers' expectations, we conducted a two day JAD (Wood & Silver 1989) (joint application design) session at Kelly AFB in order to understand what the Air Force needed, how they wanted it delivered, and to



(a) Elementary Burner Model

(b) Compound Hot Section Model

Figure 2. Elementary and Compound Models.

precisely define the application's capabilities. 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 helped to preclude proposing solutions that modified the "nail to fit the hammer".

2) 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. This insured that the complete AGETS MBR diagnostic system delivered at the end of phase two met or exceeded Air Force expectations.

3) 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 the Air Force expected and the system probably lost some acceptance due to this fact. We have learned that the performance of an AI application is one of its most important features.

Application Use and Payoff

Results

AGETS MBR has been in use by members of the San Antonio Air Logistics Center and employees of Pratt & Whitney since June of 1994. We tested AGETS MBR on the set of all telephone hotline calls received at Kelly AFB from world–wide AGETS field locations from 6/1/94 to 10/12/94. There were a total of 31 calls received. Of these calls, 13 were acceptable for AGETS MBR troubleshooting as outlined in the Section labelled 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.

Due to the conditions outlined in the AGETS MBR Section, 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 employed by AGETS MBR and to evaluate the effectiveness of AGETS MBR (*i.e.*, what the system could do if not for the limited scope of how it is routinely used), the 13 cases identified above 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 the final level of ambiguity reduction 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, and negative otherwise. AGETS MBR was able to do this in 12 of the 13 cases, leading to an overall diagnostic coverage of 92%. The other important statistic to be seen from examining the table is that the average ambiguity reduction is 80%. Ambiguity reduction is defined as the percentage of components that are removed from consideration as possibly causing the fault given the symptoms as defined in the case. These statistics are displayed graphically in Figure 3.

As of 4/15/95, a total of 77 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.

Payoff

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 underway to relax the operational require-

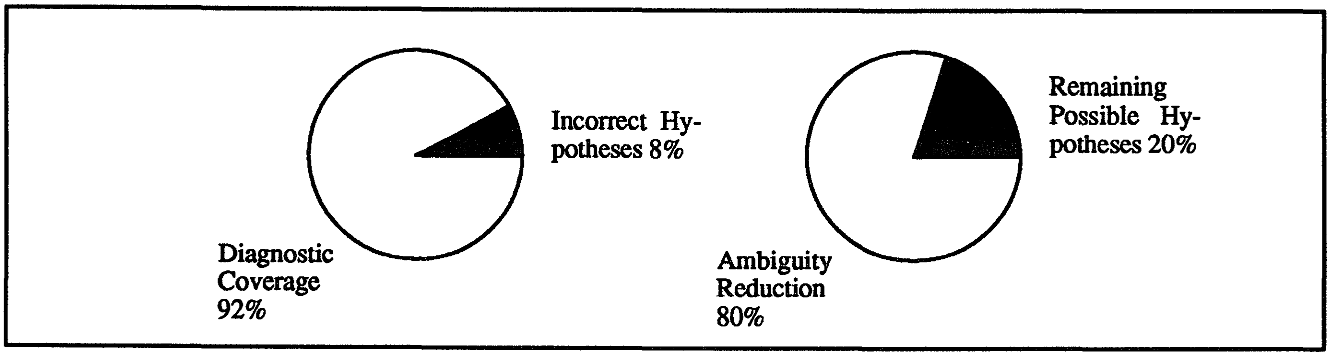


Figure 3. Diagnostic coverage and ambiguity reduction.

ments of AGETS MBR in order to diagnose to the LRU level. Post–delivery experience has shown that many difficult problems occur within AGETS. Allowing AGETS MBR to troubleshoot such problems would make payoff calculations in terms of financial and/or labor metrics more tractable.

An unexpected benefit of system delivery was the decision to base the PW development engineer at Kelly AFB 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 accomplished the goal of developing experience in artificial intelligence diagnostic systems. Specifically, an engineer acquired expertise in modeling complex systems for making accurate diagnoses. PW/GESP plans to utilize this expertise in future projects.

Application Development and Deployment

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 their domain knowledge which were reviewed by UTRC in order to ensure consistency with other models and compatibility with QRS's modeling primitives. PW then coded the models into QRS and tested their fault detection capability (*i.e.*, shallow 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 (*i.e.*, 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 man–months involving two developers at any given time (one from UTRC and one from PW). AGETS MBR encompasses three AGETS configurations (unique models are specific to a particular configuration):

- Installed, comprising 1398 elementary and 242 compound models. 212 models are unique.
- Uninstalled, comprising 1368 elementary and 229 compound models. 181 models are unique.
- Standalone, 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. The confluences were derived from the basic thermodynamic behavior of gas turbine components such as compressors, burners, and turbines. The electrical and hydromechanical control of the F100 proved to be especially challenging to model. Many iterations were required to eliminate static instability in the

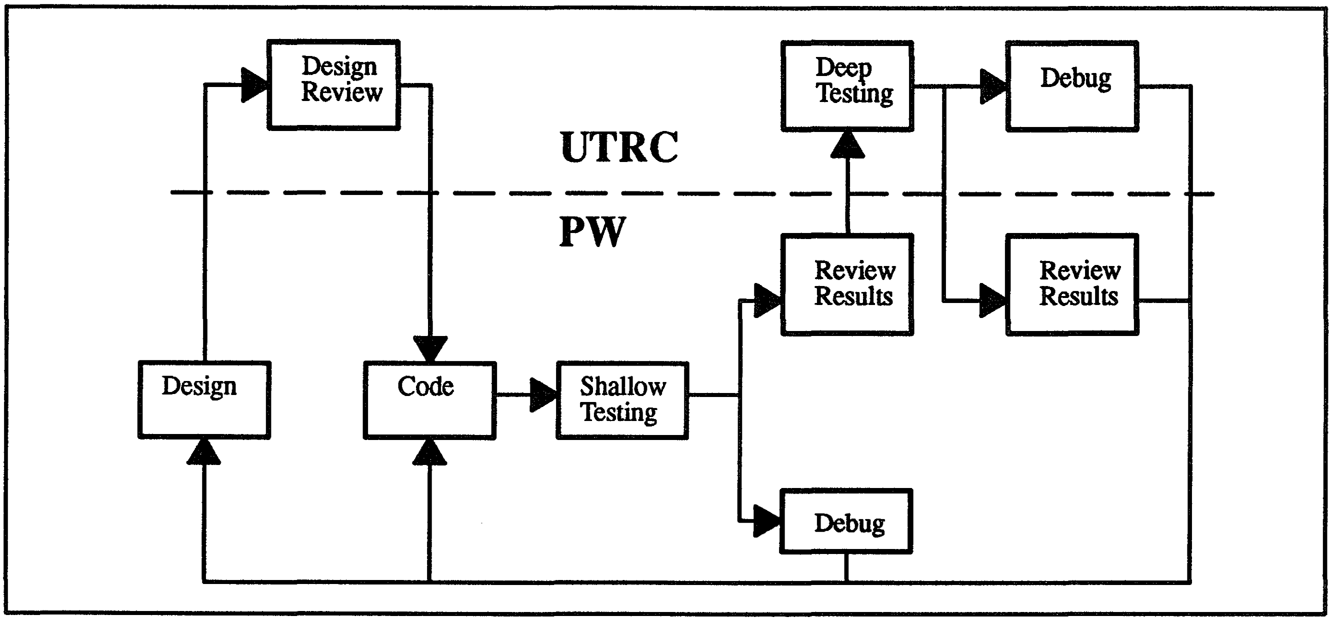


Figure 4. UTRC/PW Software Development Process.

control system model. The AGETS models were designed from electrical schematics in the AGETS maintenance manual (Organizational and Intermediate ... 1988). While 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 (*i.e.*, shallow testing), then to compile hypothesis lists (*i.e.*, deep testing). These hypothesis lists were checked against historical evidence and reviewed by domain experts. As explained above, deviations from expected results usually caused an iteration in the model design.

Test procedure design and implementation was accomplished concurrently with model testing. The majority of test procedures used by AGETS MBR were of two main types: parameter observations and 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 utilized.

The final process in the AGETS MBR development was converting the entire system from a UNIX platform, upon which it was developed, to an IBM–compatible PC platform. AGETS MBR was delivered on an IBM–compatible PC utilizing a 60 MHz Pentium processor and 64 Mb of RAM. 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 easily used by personnel at the San Antonio Air Logistics Center (SA–ALC) at Kelly AFB. It was designed to be user–friendly by incorporating an easy to use graphical user interface along with a full–featured on–line 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 of 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's manuals and conducting tutorial sessions. After completion of training, an employee of Pratt & Whitney who could answer questions about AGETS MBR remained on–site at SA–ALC.

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%
P074004 n	Yes	101	259	61%
P074004 p	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.

Maintenance

As of this writing, there have been two updates to the AGETS MBR software. The first update involved miscellaneous user–interface improvements while the second update involved a major performance improvement. Neither of these updates were 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 (*e.g.*, engine fuel system, oil system, etc.), inclusion of decision trees automatically compiled from AGETS MBR models for more efficient performance, as well as an extended tutorial system, and field deployment of the AGETS MBR software.

Pratt & Whitney 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 will be required to the knowledge base. This is in contrast to fault–based approaches (*e.g.*, fault trees or shallow rule–based systems) where new failure modes can be frequently discovered.

However, when changes to the knowledge base are needed, due to incorrect models of device behavior or changes to the actual system, updating the application should be fairly straightforward in that one qualitative model can be easily 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 very difficult to troubleshoot due to any or all of the following:

- the volume and complexity of AGETS measurements,
- possible interference by the engine's electrical and/or hydromechanical control systems, and
- the lack of a formal troubleshooting aid for the testing system.

AGETS MBR, utilizing QRS software, provides support personnel at Kelly AFB with a diagnostic aid which 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 employed in AGETS MBR.

Acknowledgements

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, 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 (Pratt & Whitney) moderated the early JAD session at Kelly AFB.

References

Clark, R.T.; Gallo, S.; and Hamilton, T.P. January 1994. Qualitative Reasoning: Theory and Applications. In Proceedings of the 32nd AIAA Aerospace Sciences Meeting. Reno, Nevada.

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

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

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

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

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

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

Hamilton, T.P. July 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. December 1984. Commonsense Reasoning about Causality: Deriving Behavior from Structure. *Artificial Intelligence* 24:169–203.

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

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

Intermediate Troubleshooting and Engine Subsystem Description, Aircraft Engine. USAF Model F100–PW–100(3). July 1984. Technical Order, T.O. 2J–F100–21–3, WP 052.

Organizational and Intermediate Maintenance Automated Ground Engine Test Set PWA52105. August 1988. Technical Order, T.O. 33D4–6–690–2.

Winston, H.; Sirag, D.; Hamilton, T.; Smith, H.; Simmons, D.; and Ma, P. January 1991. Integrating Numeric and Symbolic Processing for Gaspath Maintenance. In Proceedings of the 29th AIAA Aerospace Sciences Meeting. Reno, Nevada.

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