An Integrated Diagnostics Virtual Test Bench for Life Cycle Support

Kevin Cavanaugh
Qualtech Systems, Inc. 100 Great Meadow Road
Wethersfield, CT 06109
(860) 257-8014 x107
Kevin@teamqsi.com

Abstract - Qualtech Systems, Inc. (QSI) has developed an architecture that utilizes the existing TEAMS (Testability Engineering and Maintenance Systems) integrated tool set as the foundation to a computing environment for modeling and rigorous design analysis. This architecture is called a Virtual test Bench (VTB) for Integrated Diagnostics. The VTB approach addresses design for testability, safety, and risk reduction because it provides an engineering environment to develop/provide:

1. Accurate, comprehensive, and graphical model based failure mode, effects and diagnostic analysis to understand failure modes, their propagation, effects, and ability of diagnostics to address these failure modes.

2. Optimization of diagnostic methods and test sequencing supporting the development of an effective mix of diagnostic methods.

3. Seamless integration from analysis, to run-time implementation, to maintenance process and life cycle support.

4. A collaborative, widely distributed engineering environment to "ring-out" the design before it is built and flown.

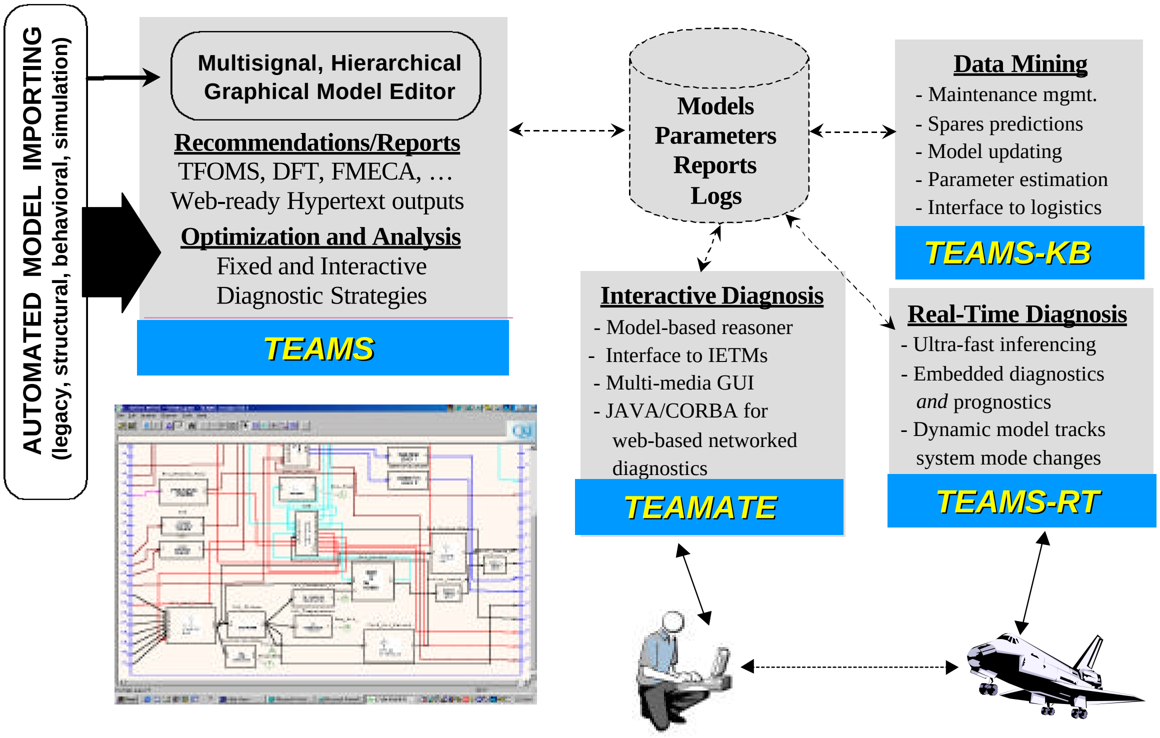
The VTB architecture offers an innovative solution in a COTS package for system/component modeling, design for safety, failure mode/effect analysis, testability engineering, and rigorous integration/testing of the IVHM (Integrated Vehicle Health Management) function with the rest of the vehicle. The VTB approach described in this paper will use the TEAMS software tool to generate detailed, accurate "failure" models of the design, assess the propagation of the failure mode effects, and determine the impact on safety, mission and support costs. It will generate FMECA, mission reliability assessments, incorporate the diagnostic and prognostic test designs, and perform testability analysis. Diagnostic functions of the VTB include fault detection and isolation metrics, undetected fault lists, ambiguity group lists, and optimized diagnostic trees.

TABLE OF CONTENTS

1. INTRODUCTION - THE QSI VTB

The QSI VTB for Integrated Diagnostics is based upon the commercially available TEAMS^{TM} tool set depicted in figure 1. The VTB provides system/component modeling, design for safety, failure mode analysis, testability engineering, and rigorous integration/testing of the IVHM (Integrated Vehicle Health Management) function with the rest of the vehicle. The VTB will be a significant step toward designing safe, robust, and highly supportable systems.

A comprehensive advanced VTB, as shown in figure 2, provides a powerful capability that will revolutionize the way aerospace engineers (or designers of any complex system) design systems for safety, reliability, maintainability, and testability. This technology leverages 10 years of work in the aerospace industry on projects with Sikorsky, NASA, Army Research Office, Boeing, Lockheed, and the Navy Surface Warfare Center. QSI has designed TEAMS-RT, TEAMATE, and TEAMS-KB to run over the web with a browser. This VTB that can be exercised, viewed, and demonstrated from any authorized location and on any computer with a web browser.


AUTOMATED MODEL IMPORTING
(legacy, structural, behavioral, simulation)
Multisignal, Hierarchical Graphical Model Editor
Recommendations/Reports
TFOMS, DFT, FMECA, …
Web-ready Hypertext outputs
Optimization and Analysis
Fixed and Interactive Diagnostic Strategies
TEAMS
Models
Parameters
Reports
Logs
Data Mining
- Maintenance mgmt.
- Spares predictions
- Model updating
- Parameter estimation
- Interface to logistics
TEAMS-KB
Interactive Diagnosis
- Model-based reasoner
- Interface to IETMs
- Multi-media GUI
- JAVA/CORBA for web-based networked diagnostics
TEAMATE
Real-Time Diagnosis
- Ultra-fast inferencing
- Embedded diagnostics and prognostics
- Dynamic model tracks system mode changes
TEAMS-RT

2. TEAMS MODELING TO SUPPORT EFFECTIVE DESIGN FOR SAFETY, DIAGNOSTICS AND PROGNOSTICS.

The ability to model and predict the "failure behavior" and resulting effect on the system is first and foremost in the development of fail-safe, fault tolerant, and effective diagnostic/prognostic capability. The engineer must be able to see how the system will fail and then determine how to address the failure. For a given failure mode, the engineer must be given tools to help optimize the diagnostic and

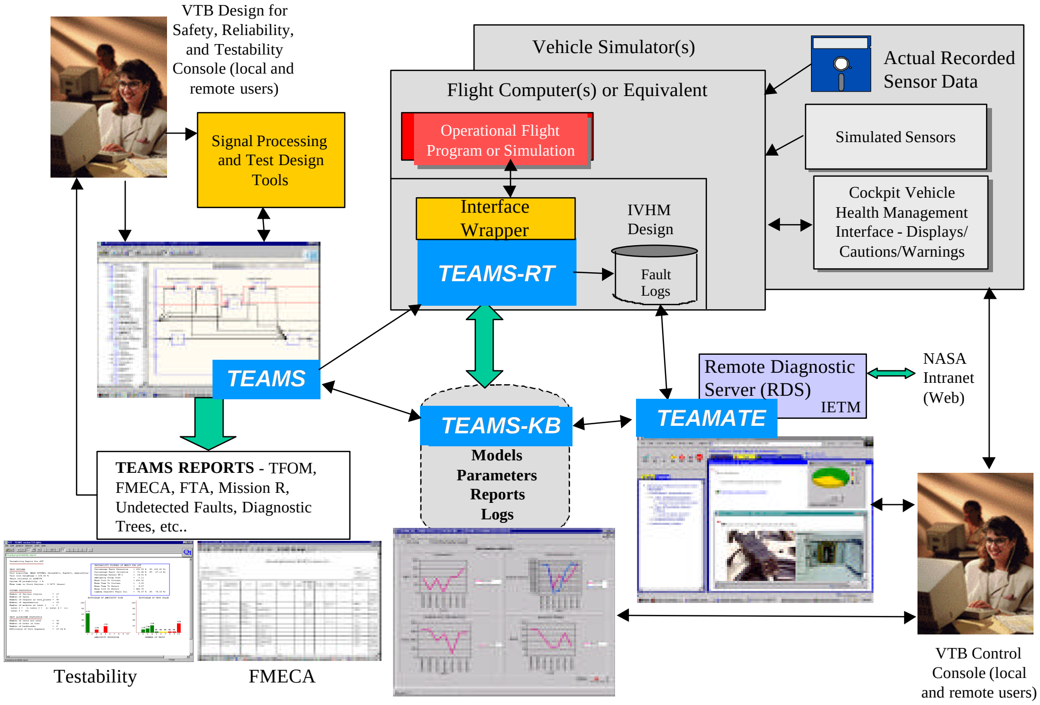

VTB Design for Safety, Reliability, and Testability Console (local and remote users)
Signal Processing and Test Design Tools
TEAMS
TEAMS REPORTS - TFOM, FMECA, FTA, Mission R, Undetected Faults, Diagnostic Trees, etc..
Testability
FMECA
Vehicle Simulator(s)
Actual Recorded Sensor Data
Flight Computer(s) or Equivalent
Operational Flight Program or Simulation
Simulated Sensors
Interface Wrapper
IVHM Design
TEAMS-RT
Fault Logs
Cockpit Vehicle Health Management Interface - Displays/ Cautions/Warnings
TEAMS-KB
Models
Parameters
Reports
Logs
Remote Diagnostic Server (RDS)
IETM
NASA Intranet (Web)
TEAMATE
VTB Control Console (local and remote users)

Figure 2 – The VTB Architecture

prognostic approach within the specification for safety, reliability, weight, costs, etc. The TEAMS^{TM} tool (figure 3) will form the foundation for this aspect of the VTB.

The original version of Qualtech Systems' Testability Engineering tool, TEAMS^{TM}, employed dependency modeling, albeit in a hierarchical directed graph format, to model systems. However, its limitation in modeling and validating large complex systems was apparent. Consequently, QSI introduced the multisignal modeling approach in 1994, which allowed the modeler to capture system information more naturally in a colored di-graph format that retained close relationship to the structure or information flow in a system. Since then, we have developed additional tools to expand the role of TEAMS^{TM} and multisignal modeling[1] beyond that of testability engineering. Qualtech has developed a complete solution for Integrated Diagnostics (ID) that addresses all the facets of Design for Safety, Testability (DFT), Reliability Analysis, Failure Modes, Effects and Criticality Analysis (FMECA), and maintenance.

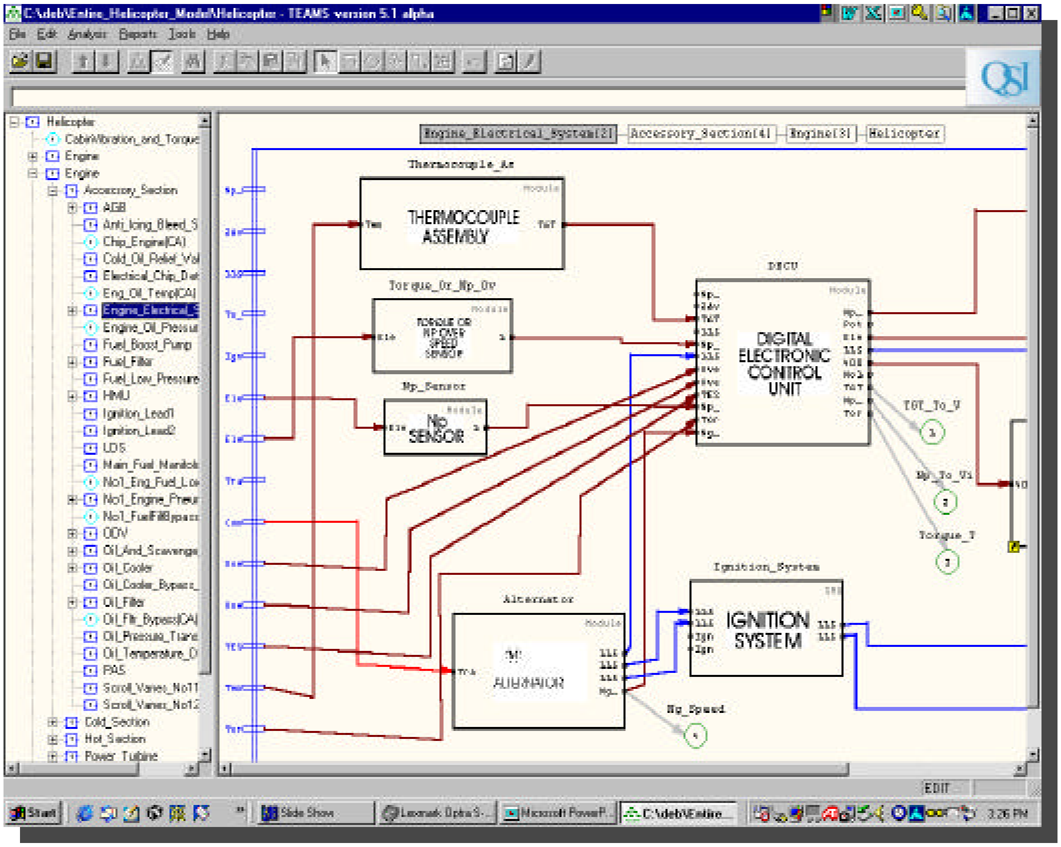


Figure 3 – Part of a TEAMS Model of a Jet Engine

1.1 *TEAMS Multisignal Modeling**

Design for Safety, Reliability, Testability and minimizing the life-cycle cost of a complex system requires a well-coordinated effort involving people of different expertise. In effect, the model is the means by which people document and convey their understanding of the system, as it relates to their fields of expertise. For example, to the design engineer, the model could be a block diagram with transfer functions, to the safety and reliability engineer it represents a block diagram of the components that can fail along with their effects. However, to a maintenance engineer, it is the schematic of replaceable components that make up the system. The objective is to develop a modeling methodology that is simple and intuitive enough so that people of various disciplines can understand and relate to it, yet powerful enough to be used during the entire life-cycle of a system.

1.2 *Observations inspiring multisignal modeling*

The foundation of multisignal modeling is based on the following observations:

First, for safety, reliability, and diagnostic purposes, we only need to model how a fault (or cause) propagates through the system and to the various monitoring points. The objective is not design verification: we assume that the system normally

[1] This work was supported in part by NASA ARC Research (contract NAS2-14320).

works to specification. The failure of one or more components (causes) results in system malfunctions (effects) that are observable at various points (test points) in the system. For FMECA, the goal is to trace the effects of a failure and assess its impact on system performance. For DFT, the goal is to ensure that the system is sufficiently observable (and controllable) so that the cause of a malfunction can be easily identified and mitigated. In field maintenance, the goal is to accurately identify the cause of a malfunction in minimum time/cost. In all these cases, it is sufficient to model the system in its failure space. Thus, the system can be modeled in terms of first-order cause-effect dependencies, i.e., how a faulty node affects its immediate neighbors. Higher-order dependencies can be inferred from first-order dependencies.

Second, the failure space is not binary (i.e., simple pass/fail), as is assumed in structural and traditional dependency models. The function space is multidimensional. Consequently, the failure space, which is the complement of function space, is also multidimensional. For example, the function of a sine wave generator is to generate a sine wave of specified amplitude, phase and frequency. It is said to have failed if the output sine wave does not have the desired amplitude, phase or frequency.

Third, since the failure state can be arbitrary, it is unnecessary to model the exact quantitative relationships. In order to illustrate this assertion, consider a cascade of three amplifiers, having gains of 2, 3, and 4, with an overall gain of 24. If, due to a fault, the new gain is 12, the first stage, with a design gain of 2, should not necessarily be implicated; the gain of any of the stages may have been reduced by half due to a failure. Thus, when the same attribute is modified by multiple components, quantitative relationships convey little, if any, information. If the gain is off, the amplifiers will be the likely suspects. So, it is only necessary to identify the important functional attributes (or the dimensions of the function space) and associate them with the appropriate components and tests. These attributes are the signals.

Fourth, there can be two distinct types of failures: *functional* failures and *general* failures. Consider a lossless (passive) filter consisting of an inductor and a capacitor. If a fault in the inductor or capacitor causes a deviation in the center frequency, it is considered a functional failure, i.e., a fault that affects the function it was supposed to perform. On the other hand, if the fault is a short-circuit that causes the output power to be zero, this is a general failure, that is, a catastrophic failure affecting attributes beyond its normal functioning by interrupting the flow of information through it. Thus, a failure in a module can either affect the attributes it was supposed to (functionally) modify, or all the attributes flowing through it. This affects how the overall cause-effect dependencies are derived from the structure and signal information

1.3 Basic constructs in multisignal modeling

Multisignal modeling methodology is a hierarchical modeling methodology, where the propagation paths of the effects of a failure are captured in terms of a directed graph. The graph has four different kinds of nodes:

- The **Module** corresponds to a piece of hardware with a certain set of functions (captured in terms of signals). Modules themselves can be described in terms of another graph consisting of (sub)modules and other nodes - allowing for hierarchical modeling. A module at the lowest level is called a *failure mode* or an *aspect* or an *anomaly*. Modules are the nodes that fail, diagnosis being the process of identifying the failure source(s) from test results.
- The **Test Point** corresponds to locations (Physical or logical) where measurements can be made. A test point can have multiple tests **-** i.e., at a single physical location (or probe point) where one or more measurements may be made. Such tests can be classified as safety tests, performance tests and diagnostic tests, as is common in TPS development, or can be associated with *levels* to model different echelons of maintenance. TEAMS can also include information regarding setup operations that need to be performed and resources that are needed to perform a certain test, and can optimize the diagnostic strategy subject to these constraints.
- The **AND** node captures redundancy information. For example, if both A and B has to fail, before C is affected, A and B will be connected to C via an AND node. AND nodes allow us to model fault-tolerant systems for diagnosis and reliability and criticality analysis.
- The **SWITCH** node captures conditional connections or change in interconnections due to model changes. Switches let us model dynamic and reactive systems.

These nodes are interconnected using links, forming a hierarchical graph. Propagation algorithms convert this graph to a single global fault dictionary (or D-matrix), for a given mode and state of the system. This D-matrix contains the basic information needed to interpret test results and diagnose failures (onboard monitoring), and generate optimized test sequence that minimizes the troubleshooting time (field maintenance).

1.4 Advanced multisignal modeling

Advanced features in the TEAMS^{TM} toolset include:

- <u>Signal grouping</u>: A designer makes up a complex function out of simple functions. Similar capability of grouping low-level signals to form a high-level “super-signal” is provided. As an example, harmonic distortion, signal-to-noise ratio, linearity, etc. can all be encompassed by one “super-signal” called fidelity.
- <u>Signal aliasing</u>: Since it is conceivable that different groups of people from diverse disciplines will use varied terminology to refer to the same function or signal,

signal synonyms or aliases will be necessary for integration of multisignal models.

- Signal blockers: Signal blockers provide barriers to propagation of certain signals. For example, the 1553 bus system uses multiple bus couplers which buffers the d.c. biases and loading effects of a catastrophic failure. This was modeled using signal blockers for all d.c. signals (resistance, current, voltage, etc.) and the general failure
- Signal mappers: Signal mappers are used to model transducers that transform one signal to another. For example, a speaker transforms an electrical signal to sound waves, while maintaining the information content and characteristics (e.g., noise, distortion).

1.5 Simple Guide to multisignal modeling

In the following, we provide a three-step procedure for multisignal modeling that should be adequate for most modeling needs:

1. *Enter the structural model, schematic model or a conceptual block diagram.* In TEAMS™, the structural model can be automatically generated from structural models or netlists (e.g., VHDL, EDIF), or directly entered via the graphical user interface. TEAMS™ can also import XML files (formatted to the TEAMS dtd).
2. *Add signals to the modules and test points*. The set of signals can be identified from the functional specification or from the independent variables in the transfer function (e.g., the signal specification of a power amplifier could include output distortion, harmonic distortion and power output). In general, any unique attribute will have an associated signal.
3. *Update models with additional information*. For example,
   - identify and model the redundant components using AND nodes.
   - identify and model modes of operations using SWITCHs.
   - provide additional test information, such as setup operations, resource requirements, confidence, diagnostic run levels, etc.
   - identify signal blockers, mappers, and group signals for clarity.
4. *Validate the model.* This is a critical step, since the analysis results can only be as good as the models. In TEAMS™, the users can interactively seed faults and identify affected tests and vice-versa. Peer review and actual integration with run-time tools (i.e., TEAMATE and TEAMS-RT) also provide invaluable feedback on the accuracy of the model.

3. FMECA AND DESIGN FOR SAFETY ANALYSIS USING MULTISIGNAL MODELS

The multisignal models capture the following information necessary for the automation of Failure Modes, Effects, and Criticality Analysis:

- The *failure modes* (i.e., aspects of anomalies)
- The reliability (MTTF or failure rate) of each component

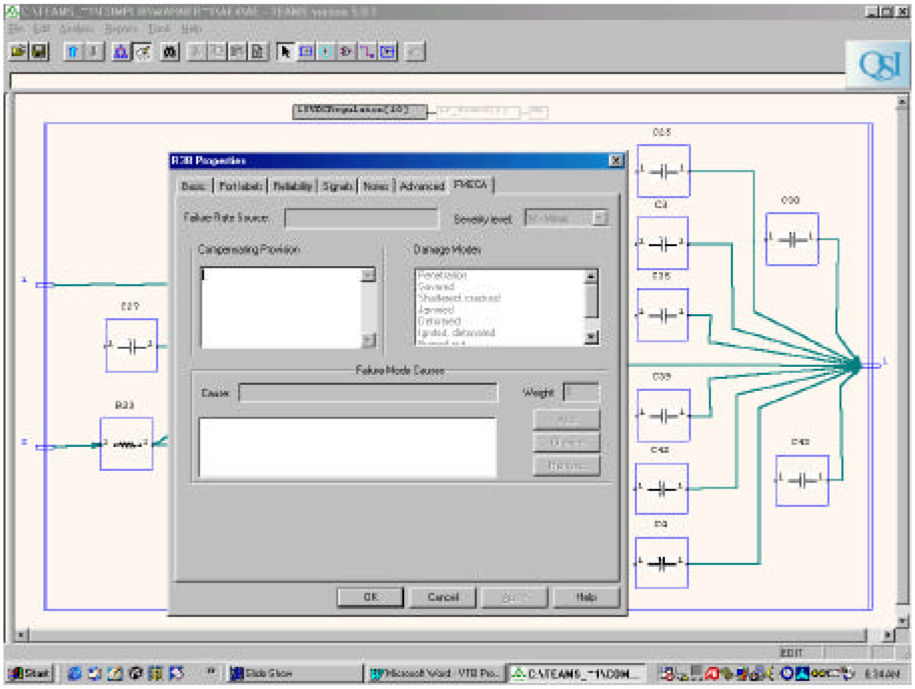
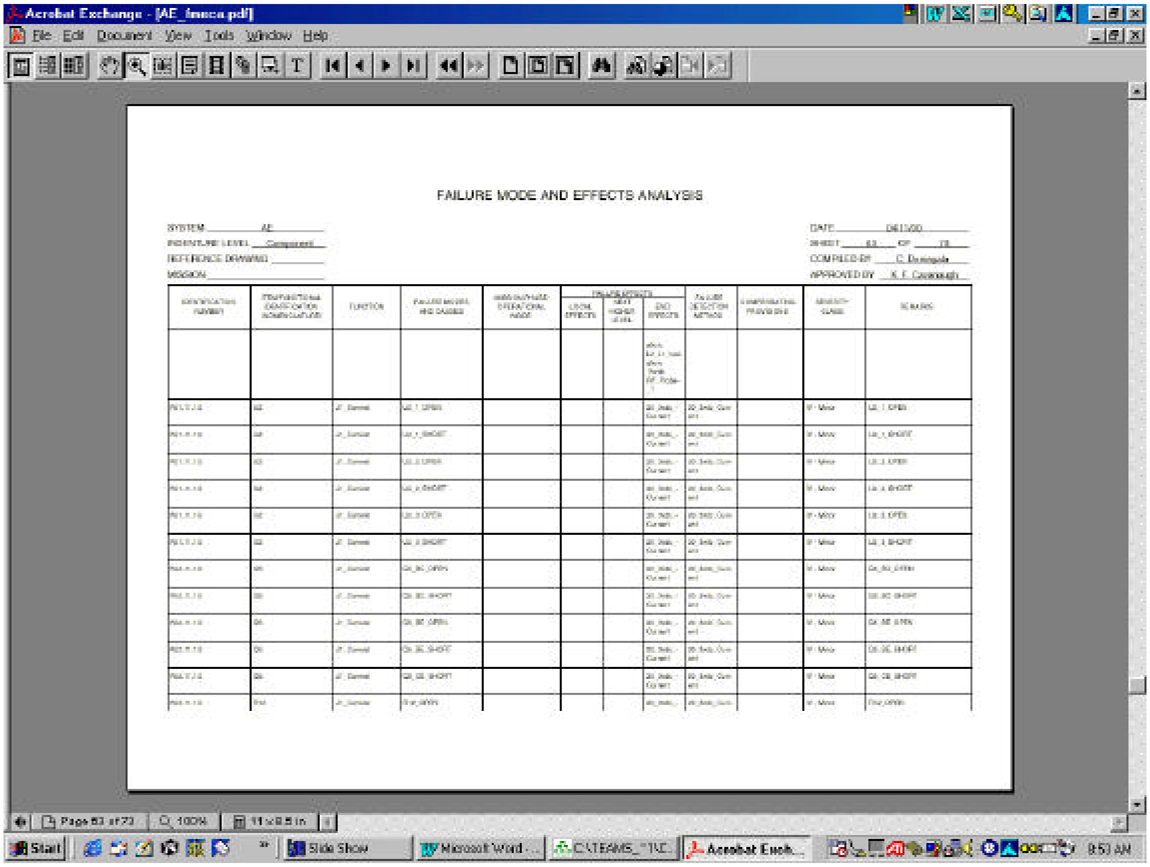


Figure 4 – TEAMS FMECA Input Panel

- The component and test hierarchy, and hence the *Indenture Level* for the FMECA analysis
- The failure *criticality* of each component
- Elementary *functions* performed by each component via the signals attached to each component
- *Effects* of the failures of components, in terms of signals (or measurements) associated with the tests that detect them
- Redundancies in the system modeled via AND nodes which allow M-out-of-N switching logic, used to compute whether a failure effect impacts system safety, performance or is masked by redundancy
- The connectivity of components that helps establish cause-effect relationships
- The Phases of operations for a mission in FMECA analysis is equivalent to the system modes in TEAMS multisignal models
- Additionally, TEAMS™ can also generate the diagnostic path to identify the particular failure mode.



FAILURE MODE AND EFFECTS ANALYSIS

SYSTEM AE
INDENTURE LEVEL Component
REFERENCE DRAWING
MISSION

DATE 061500
SHEET 63 OF 73
COMPILED BY C. Domagala
APPROVED BY K. F. Cavanaugh

IDENTIFICATION NUMBER	ITEM/FUNCTIONAL IDENTIFICATION (NOMENCLATURE)	FUNCTION	FAILURE MODES AND CAUSES	MISSION PHASE/ OPERATIONAL MODE	FAILURE EFFECTS			FAILURE DETECTION METHOD	COMPENSATING PROVISIONS	SEVERITY CLASS	REMARKS
					LOCAL EFFECTS	NEXT HIGHER LEVEL	END EFFECTS				
							[illegible]				
[illegible]	[illegible]	[illegible]	LU_1 OPEN				[illegible]	[illegible]		IV - Minor	LU_1 OPEN
[illegible]	[illegible]	[illegible]	LU_1_SHORT				[illegible]	[illegible]		IV - Minor	LU_1_SHORT
[illegible]	[illegible]	[illegible]	LU_2 OPEN				[illegible]	[illegible]		IV - Minor	LU_2 OPEN
[illegible]	[illegible]	[illegible]	LU_2_SHORT				[illegible]	[illegible]		IV - Minor	LU_2 SHORT
[illegible]	[illegible]	[illegible]	LU_3 OPEN				[illegible]	[illegible]		IV - Minor	LU_3 OPEN
[illegible]	[illegible]	[illegible]	LU_3 SHORT				[illegible]	[illegible]		IV - Minor	LU_3 SHORT
[illegible]	[illegible]	[illegible]	Q4_BC_OPEN				[illegible]	[illegible]		IV - Minor	Q4_BC_OPEN
[illegible]	[illegible]	[illegible]	Q4_BC_SHORT				[illegible]	[illegible]		IV - Minor	Q4_BC_SHORT
[illegible]	[illegible]	[illegible]	Q4_BE_OPEN				[illegible]	[illegible]		IV - Minor	Q4_BE_OPEN
[illegible]	[illegible]	[illegible]	Q4_BE_SHORT				[illegible]	[illegible]		IV - Minor	Q4_BE_SHORT
[illegible]	[illegible]	[illegible]	Q4_CE_SHORT				[illegible]	[illegible]		IV - Minor	Q4_CE_SHORT
[illegible]	[illegible]	[illegible]	[illegible]_OPEN				[illegible]	[illegible]		IV - Minor	[illegible]_OPEN

Figure 5 – Example TEAMS Generated FMECA Report in MIL STD Format

Thus, the multisignal models capture sufficient information to substantially automate the Failure Modes, Effects and Criticality Analysis and the associated design for safety assessment. New features have been added to TEAMS to automate the propagation of signals through the model based on the established links. This feature gives the engineer insight into the effects of a failure mode at each level in the model hierarchy. As part of a VTB customization effort QSI can develop hooks to existing FMECA libraries and reliability databases such as those published by Rome Air Center. The information in the model could also be presented in Fault Tree Analysis (FTA) format. As part of the VTB development/customization, QSI can add these capabilities to the TEAMS™ tool. Figure 4 shows the FMECA input panel for a component in a TEAMS model of an antenna electronics circuit board. Figure 5 is an example of a TEAMS™ generated FMECA report based on the model.

This FMECA aspect of the VTB leverages elements of our current phase II SBIR work with the Navy Surface Warfare Center, Dahlgren entitled “Failure Analysis of Large Scale Systems”. This project is developing TEAMS™ enhancements in the way an engineer “visualizes” failure modes and their effects on the system. The project also involves Human-System Interaction and the possible contribution that human error could make to system failure. The failure analysis model for this project is a command and control system. In effect, the failure analysis will cover system design and also the “process” by which it is used.

1.6 Reliability and Availability Analysis using multisignal models

Computation of reliability and availability of a system requires enumeration of all the single, double, triple,.., n-tuple failures that result in a loss of system function [7]. Clearly, such an approach has exponential complexity and consequently is infeasible for even the simplest of models with just tens of components. We, therefore, compute lower and upper bounds on the reliability of a system, using a simple, but novel, approach that is of polynomial complexity and can be applied on models with thousands of components.

Let A be the set of all faults, S be the set of faults that directly affect (i.e., singletons) system functions, and U be the set of faults that have no impact on system function - i.e., do not affect system outputs. Therefore, M = {A - U - S} is the set of redundant components, i.e., a single failure in this set does not cause a loss of function. Let Pr{X} be the probability of failure in one or more components of set X, and Pr{X ≥ 2} be the probability of 2 or more failures in X. The equivalent failure rate of all the components in set X, assuming independent Poisson arrival process, is

$$l_X = \sum_{i \in X} l_i$$

where ${}_i$ is the failure rate of component *i*, Thus, the probability of one or more failure in X at time t is

$$\Pr(X) = 1 - e^{-l_X t}$$

and, probability of two or more failures in X is

$$\Pr(X \geq 2) = 1 - e^{-l_X t} - lte^{-l_X t}$$

Thus, the worst case reliability (R) of the system is (1-Pr{A-U}), i.e., if any fault in the system with a path to the system output could bring the system down. This is a lower bound on the reliability, and is an exact expression for reliability of a system without any redundancy (i.e., when A-U=S).

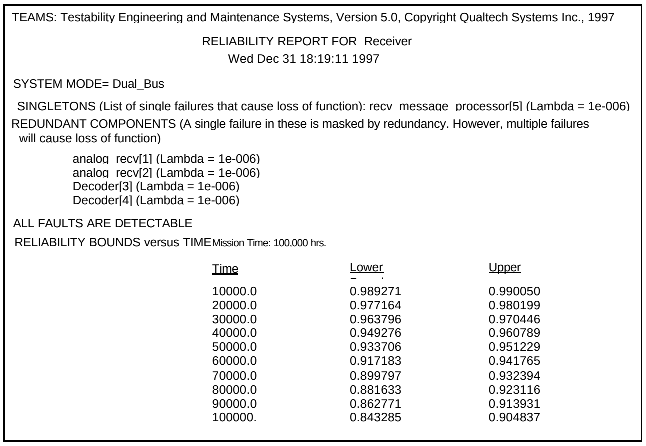
A tighter lower bound on the reliability (R) of the system is (1- Pr{S} - Pr{M^2}), i.e., if any 2 failures in M leads to loss of function. This bound can be further refined by identifying disjoint sets in M={M_1:M_2:M_3:...} that do not share any redundancy. Then the revised bound will be

$$\left(1 - \Pr\{S\} - \Pr\{M_1 \geq 2\} - \Pr\{M_2 \geq 2\} - \ldots\ldots\right)$$

The best case reliability of the system is (1-Pr{S}), i.e., if only singletons could lead to loss of functions, and the doubletons, tripletons, etc. have no significant contribution to system downtime. This is therefore an upper bound on reliability. Therefore, the reliability of a system can be bounded as:

$$1 - \Pr\{A - U\} \leq 1 - \Pr\{S\} - \Pr\{M \geq 2\} \leq R \leq 1 - \Pr\{S\}$$

A sample reliability and availability report generated from the TEAMS model is presented in Figure 6.



```
TEAMS: Testability Engineering and Maintenance Systems, Version 5.0, Copyright Qualtech Systems Inc., 1997

                         RELIABILITY REPORT FOR  Receiver
                            Wed Dec 31 18:19:11 1997

SYSTEM MODE= Dual_Bus

 SINGLETONS (List of single failures that cause loss of function): recv  message  processor[5] (Lambda = 1e-006)
REDUNDANT COMPONENTS (A single failure in these is masked by redundancy. However, multiple failures
  will cause loss of function)

          analog  recv[1] (Lambda = 1e-006)
          analog  recv[2] (Lambda = 1e-006)
          Decoder[3] (Lambda = 1e-006)
          Decoder[4] (Lambda = 1e-006)

ALL FAULTS ARE DETECTABLE
RELIABILITY BOUNDS versus TIMEMission Time: 100,000 hrs.

                    Time            Lower           Upper

                    10000.0         0.989271        0.990050
                    20000.0         0.977164        0.980199
                    30000.0         0.963796        0.970446
                    40000.0         0.949276        0.960789
                    50000.0         0.933706        0.951229
                    60000.0         0.917183        0.941765
                    70000.0         0.899797        0.932394
                    80000.0         0.881633        0.923116
                    90000.0         0.862771        0.913931
                    100000.         0.843285        0.904837
```

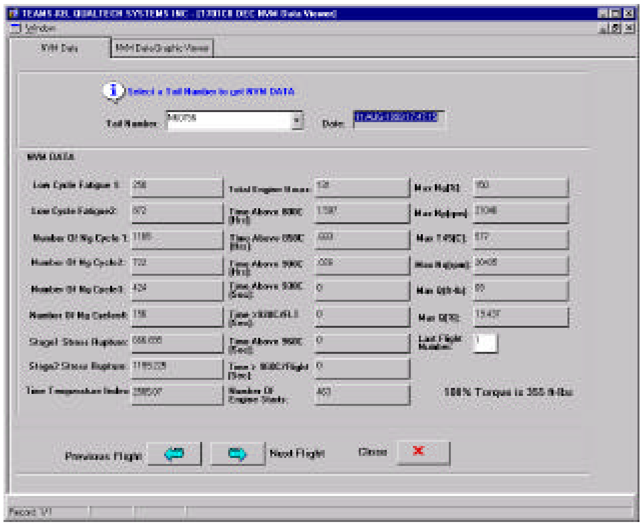
Figure 6 - A Sample Mission Reliability Report Generated from TEAMS

1.7 Model Information Management:

A good model captures the expert’s knowledge of the underlying system that is being modeled. An effective management of the information, once validated, leads to a significant reduction of the total cost of model development. This reduction of the Total Cost of Model development is a key to successful adoption of any such model development methodology. One of the important cornerstones of Model Information Management (MIM) is the development of a practical and deployable Reusable Test and Model Library (RTML). At present, QSI has a prototype Reusable Test Library (RTL) as a part of the RTML. The RTML, once implemented, will immediately incur the following benefits:

- Model development cost will be reduced since identical and similar parts are not modeled repeatedly
- A large archive of sample models will enable new users to learn the modeling methodology quickly
- The quality of the models will be improved since only the “best of class” models will be shared

TEAMS-KB stores and manages TEAMS^{TM} models and it provides capabilities to update a system’s component reliability and failure rates in response to cumulative maintenance and repair data. QSI is investigating enhancements that take into account available regime or operational-mode dependent usage data from the system and related maintenance data extracted from IMIS and Logistics databases. Figures 7 and 8 illustrate two of the several interfaces in TEAMS-KB, customized to support maintenance logbook functions for a Navy program, for interrogating diagnostic, and maintenance, and logistics information. The TEAMS-KB upgrades (new functions and algorithms), as a result of our research activity, will provide plots of life remaining against a number of maintenance/logistics parameters such as future mission reliability, spares consumption, support equipment needs, personnel needs/training, etc.



TEAMS-KB, QUALTECH SYSTEMS INC - [T701C8 DEC NVM Data Viewer]

Select a Tail Number to get NVM DATA

Tail Number: N60756 Date: [illegible]

NVM DATA

Low Cycle Fatigue 1: 258
Low Cycle Fatigue2: 872
Number Of Ng Cycle 1: 1185
Number Of Ng Cycle2: 772
Number Of Ng Cycle3: 424
Number Of Ng Cycle4: 158
Stage1 Stress Rupture: 558.658
Stage2 Stress Rupture: 1195.225
Time Temperature Index: 258507

Total Engine Hours: 128
Time Above 800C [Hrs]: 1.597
Time Above 850C [Hrs]: .603
Time Above 900C [Hrs]: .208
Time Above 930C [Sec]: 0
Time >930C/FLT [Sec]: 0
Time Above 960C [Sec]: 0
Time > 960C/Flight [Sec]: 0
Number Of Engine Starts: 463

Max Ng[%]: 150
Max Np[rpm]: 21048
Max T45[C]: 577
Max Nq[rpm]: 30408
Max Q[ft-lb]: 89
Max Q[%]: 15.437
Last Flight Number: 1

100% Torque is 355 ft-lbs

Figure 7 - TEAMS-KB: Engine performance data captured from a nonvolatile memory unit (NVM).

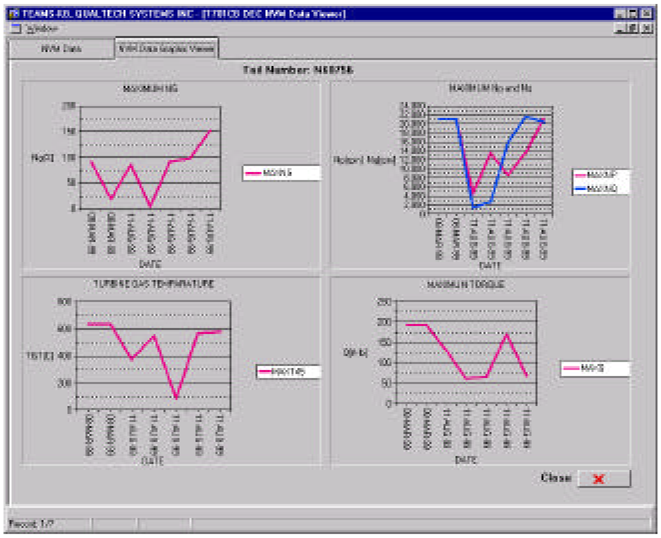


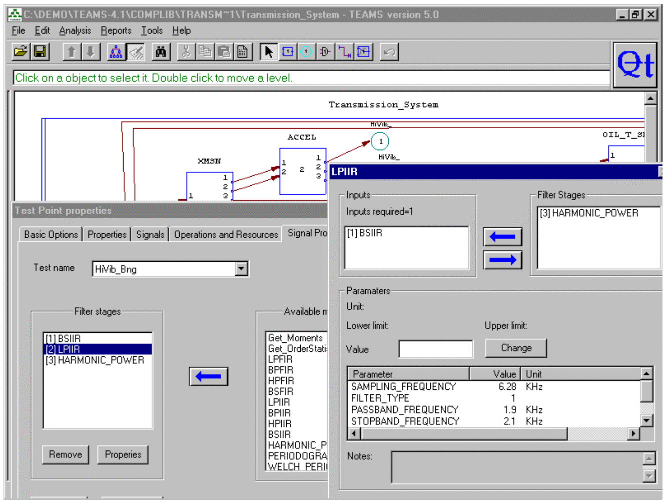
Figure 8 - TEAMS-KB: Trending analysis of a helicopter engine performance data.

In this VTB architecture, TEAMS-KB becomes the “glue” that integrates the various pieces of the system. It is also a key integration point with NASA’s Intelligent Synthesis Environment (ISE) program and other logistics tools or “logistics modeling environments”.

4. THE QUALTECH SIGNAL PROCESSING (SP) TOOL-KIT COMPANION TO TEAMS

The SP tool-kit was developed with the University of Connecticut (as a subcontractor) for basic filtering of noisy raw measurements as well as more sophisticated extraction of statistical features to support hypothesis testing and pattern recognition. This tool-kit is configured as a library of routines that can be used to define tests in a TEAMS^{TM} system model. Figure 9 shows the TEAMS^{TM} modeling environment with user panels for specifying tests in a system model using a multiple-stage cascade of processing and decision-making elements (a part of a transmission system model is shown). The routines provided in the tool-kit library have a design component for the user to specify the performance parameters of the routines, and a run-time component to implement the pre-designed processing routines. The run-time routines can be invoked by a scheduler-dispatcher wrapper integrated either with the TEAMS-RT run-time environment for onboard health monitoring, or with a TEAMATE-IETM-driven automatic test equipment (ATE).

Currently, the signal processing tool-kit contains a library of several noise filtering, basic spectral analysis, and statistical routines, as well as more specialized capabilities that include wavelets (Haar prototype), autoregressive spectral methods, and nonlinear transformations. The tool-kit is designed to be a continuously evolving library that can be customized to a target environment by simply modifying configuration files to expose the appropriate functions. Once the engineer has created the model, he/she can bring up the signal processing tool-kit and select/design diagnostic/prognostic tests and maintenance options and then “drop” them into the model. TEAMS^{TM} can then evaluate the effectiveness of the proposed diagnostic/prognostic approach.



C:\DEMO\TEAMS-4.1\COMPLIB\TRANSM~1\Transmission_System - TEAMS version 5.0

Click on a object to select it. Double click to move a level.

Test Point properties

Test name: HiVib_Bng

Filter stages:
[1] BSIIR
[2] LPIIR
[3] HARMONIC_POWER

Available m
Get_Moments
Get_OrderStati
LPFIR
BPFIR
HPFIR
BSFIR
LPIIR
BPIIR
HPIIR
BSIIR
HARMONIC_P
PERIODOGRA
WELCH_PERI

LPIIR

Inputs
Inputs required=1
[1] BSIIR

Filter Stages
[3] HARMONIC_POWER

Paramaters
Unit:
Lower limit:
Upper limit:
Value

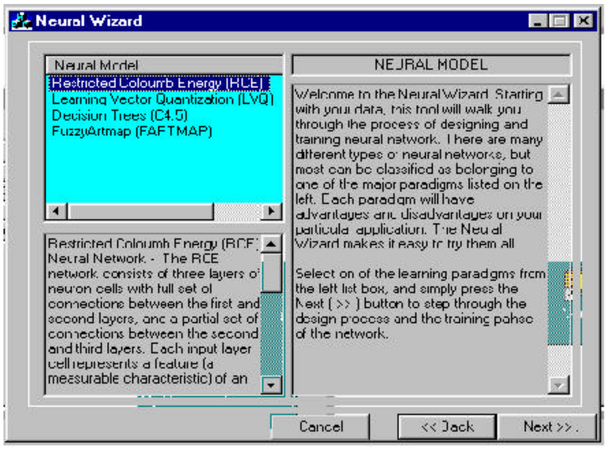
Parameter	Value	Unit
SAMPLING_FREQUENCY	6.28	KHz
FILTER_TYPE	1	
PASSBAND_FREQUENCY	1.9	KHz
STOPBAND_FREQUENCY	2.1	KHz

Notes:

Figure 9 - TEAMS interface with the Test Point properties panel and a stage Properties panel within the SP Tool-kit.

In commonly occurring cases, the detection of anomalies simply involves comparing (de-noised) scalar signals to predefined thresholds for ‘exceedances.’ In many complex systems, however, sensor signals cannot be de-linked and viewed independently of one another. In such cases, the vector of raw sensor signals (or the vector of features extracted from raw signals) must be analyzed as a pattern that can be classified into one of a set of normal and anomalous categories. Neural networks remain one of the most promising techniques for complex, non-parametric data fusion and pattern matching and are included in the tool-kit (see Figure 10). Other promising methods that are being developed include Multilayer perceptron (MLP), the Radial Basis Function (RBF), the Class Specific Classifier, and Bayesian Data Reduction.

The types of measurements and the kinds of symptoms that need to be recognized will again guide the selection of appropriate techniques. Other considerations will include the quantity of available archived 'training' data for learning/data mining, the performance of a technique in the presence of noisy data, etc. These capabilities, coupled with the FMECA functionality, provide a very powerful engineering tool.



Restricted Coloumb Energy (RCE) Neural Network. - The RCE network consists of three layers of neuron cells with full set of connections between the first and second layers, and a partial set of connections between the second and third layers. Each input layer cell represents a feature (a measurable characteristic) of an

Welcome to the Neural Wizard. Starting with your data, this tool will walk you through the process of designing and training neural network. There are many different types of neural networks, but most can be classified as belonging to one of the major paradigms listed on the left. Each paradigm will have advantages and disadvantages on your particular application. The Neural Wizard makes it easy to try them all

Select on of the learning paradgms from the left list box, and simply press the Next (>>) button to step through the design process and the training pahse of the network.

Figure 10 - A prototype neural network design environment within the SP Tool-kit.

5. Embedded Run-Time Failure Handling, Diagnostics and Prognostics.

Once the models have been completed in TEAMS$^{\text{TM}}$, and the monitoring approach optimized for reliability, diagnostics and prognostics, the system model along with it's test specifications (diagnostics/prognostics) can be exported to the TEAMS-RT ("RT" stands for "run-time") software tool. This is a "seamless" process within the integrated tool set. The TEAMS-RT based reasoning engine can efficiently process pass/fail outcomes of thousands of tests in a fraction of a second to assess the health of the vehicle. It will, therefore, form the heart of our on-board diagnostics/prognostics health management solution by "**fusing**" vehicle monitoring data to accurately detect, and isolate failed or failing components. The TEAMS-RT based "run-time" software module can be embedded in the VTB run-time environment on the actual flight computer or equivalent. It, however, needs test decisions to assess system health. Thus, raw sensor data needs to be processed and converted into the binary pass/fail format required by the diagnostic engine. A typical schematic of the preprocessing module for a local subsystem would be as shown in Figure 11. The task of the first stage would be to minimize the effects of noise by suitable filtering. The next stage would perform functional and heuristic crosschecks to exploit hardware and functional redundancies in the system to reduce, and possibly eliminate, erroneous, contradictory or duplicate sensor results. The third stage would be essentially that of hypothesis testing, and, depending on the nature of signal(s) received from the sensor(s), would be of one of possibly three types. A range-checking sub-module would process data from a non-dynamic source. In cases where the sensor signal is complex, model-based reasoning may be required to evaluate it for possible faults. Finally, the trend analysis and prognostics sub-module would monitor slow degradation in sensor signals. The resultant test decisions are processed by TEAMS-RT in real-time to assess system health, i.e., identify healthy components and isolate failing and failed components. The architecture shall be data driven and configured by system models and test scripts. The modular signal processing library will enable us to easily add new tests and provide us with a flexible and versatile means of implementing a variety of test procedures and prognosis and trending algorithms.

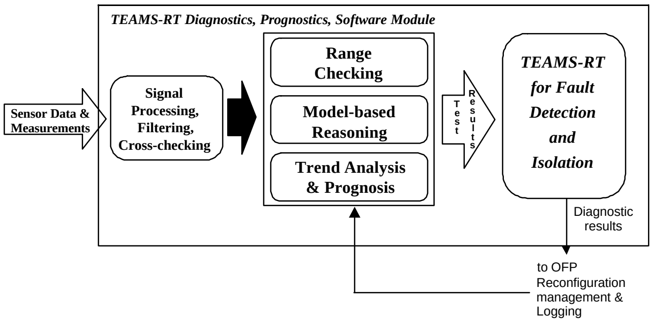

TEAMS-RT Diagnostics, Prognostics, Software Module
Sensor Data & Measurements
Signal Processing, Filtering, Cross-checking
Range Checking
Model-based Reasoning
Trend Analysis & Prognosis
Test Results
TEAMS-RT for Fault Detection and Isolation
Diagnostic results
to OFP Reconfiguration management & Logging

Figure 11 – The TEAMS-RT Embedded Software Module

The objective of the TEAMS-RT inference engine is to associate one of four distinct (failure) states with each component in the system: (1) *Good*, (2) *Bad*, (3) *Suspected*, and (4) *Unknown*. When TEAMS-RT is invoked, we assume that the state of all components is *Unknown*. If a test covering a component passes, its state is updated to *Good*. If a test covering a component fails, its state is *Suspected*. The *Bad* components are derived from these *Suspected* components by elimination of *Good* components. Many systems flying today assume that components are good until a test indicates failure. The TEAMS-RT approach considers all test conditions (pass, fail, unknown) to detect and isolate faults. TEAMS-RT does not assume a component is good just because a test has not identified it as failed. This approach results in a much more conservative (safe) and accurate diagnosis.

1.8 Capabilities and Performance of TEAMS-RT

The production version of TEAMS-RT includes additional capabilities for dynamic system mode changes, and capability of diagnosis and prognosis in fault-tolerant systems with built-in redundancy. Some unique features of TEAMS-RT are: (i) efficient real-time processing of sensor results, (ii) update of fault - test-point dependencies in response to system mode changes, and (iii) update of dependencies resulting from failures in redundant components. Table 1 presents test results for TEAMS-RT running on a distributed network on several different systems:

Model	Number of Tests Pass / Fail	Number of Faults inserted / total modeled	RT-agent CPU run time in ms (incl. TEAMS-RT)	Broker CPU run time in ms	TEAMS-RT CPU run time in ms
1553	59 / 2	2 / 174	60	3	< 5
Transmission system	46 / 5	2 / 160	54	3	< 5
EEATCS	9 / 134	2 / 78	60	3	< 5
Documatch	175 / 5	2 / 259	65	3	< 5
LO2	329 / 39	3 / 167	85	3	< 5
Engine System	274 / 32	3 / 255	85	4	< 10
LGCU-WRA	1003 / 316	4 / 2080	600	5	< 250

Table 1 – TEAMS-RT Performance Metrics

1.9 SIGNAL Processing Front-End

While TEAMS-RT was designed to be an embedded diagnostic engine for real-time systems, its applicability depends on the means for inputting binary data such as the pass/fail outcomes of tests, and the occurrence or nonoccurrence of symptoms, alarms or exceptional events. This need motivated the creation of a front-end in the TEAMS-RT architecture which can allow systems integrators to include both fault detection and fault isolation into the system monitoring environment. Since the fault detection strategy can vary widely depending on the nature of the system and the kind of sensor measurements received from the system, the architecture must allow integration of a wide range of detection strategies without the need for creating different versions of the environment for different applications.

The architectural enhancements allowing for a data preprocessing front-end to TEAMS-RT have achieved this by separating the monitoring module from the diagnostics module. The monitoring module uses a library of available preprocessing methods from which the appropriate method(s) can be chosen for a given application. The library, on the other hand, can be made richer to broaden the domain of application of the TEAMS-RT environment without the diagnostic or monitoring engines needing to be modified or recompiled.

The preprocessing module was designed with three environments/stages in view:

1. Specification environment: The user enters the design specifications for the preprocessing front-ends to all test-points in the TEAMS^{TM} model.
2. Preprocessing module design environment: The design specifications are used to synthesize the processing stages, test their functionality, and fine-tune the design of the processing stages.
3. Runtime environment: The synthesized preprocessing front-end designs are used by TEAMS-RT in the deployed system to process real-life data in real time to generate test outcomes.

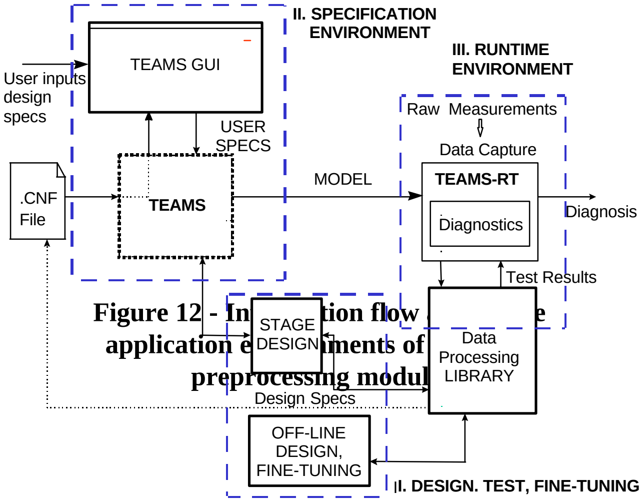


Figure 12 - In tion flow e application e ments of preprocessing modul

The data flow among the three environments is shown in Fig. 12. This scheme was used to create a library of signal-processing modules to process sensor signals from rotorcraft machinery.

The preprocessing routines can be accessed from TEAMS^{TM} through the test point properties panel (see section 4). The user can build the signal processing front-end at the test-point by specifying the number of signal-processing stages, their names, and their interconnections. In addition, the user can enter the design parameters specific to each stage. The properties that the interface prompts the user to enter are read from a configuration file that lists the specifications of each module in the library. As new preprocessing methods are added to the library, their corresponding entries are added to the configuration file. The desirable feature of this design approach is that the TEAMS-RT software does not have to be modified as the library grows.

For applicability to aircraft and rotorcraft machinery, a signal-processing library was created. The following list is a sample of routines that have been incorporated:

1. Statistics
   - Univariate moments: mean, standard deviation, skewness, and kurtosis
   - Order statistics
2. Filtering

- FIR filtering: low-pass, high-pass, band-pass, and band-stop
- IIR filtering: low-pass, high-pass, band-pass, and band-stop

3. Spectral Analysis
   - FFT, zero-padded
   - Dominant frequency
   - Harmonic powers
   - Frequency-sweep parameters
   - Periodogram
   - Welch spectrum
   - Overlapped sections, windowing
4. Nonlinearities: logarithm, polynomial, soft-limiter
5. Tests: threshold, CUSUM

For example, a preprocessing front-end for detecting a problem in a gear-box could be implemented to monitor the gear-shaft vibration signal. The modules in this front-end could contain a cascade of suitable noise-rejection filters (band-pass, high-pass, etc.), followed by a module for measuring harmonic powers in the vibration signal, and thresholding the power in the chosen harmonic(s). The output of this cascade of signal-processing stages would be a binary signal that indicates whether or not the harmonics in the vibration signal are unacceptable, denoting, in turn, a problem in the gear system such as a chipped gear-tooth. The SP tool-kit and processing libraries will continue to grow as more diagnostics and prognostics methods/procedures are developed and added by the domain experts.

6. Maintenance Process/Procedure Design and Analysis on the VTB

This approach incorporates the client's Interactive Electronic Technical Manual (IETM) and TEAMATE reasoner in the VTB for verification of maintenance procedures and accuracy of diagnostics logic for maintenance. Qualtech believes that a comprehensive "Design for Safety" and "Design for Supportability" environment must include the ability to design and evaluate the maintenance process. There have been many cases where inadequate maintenance procedure/process has led to catastrophic results. Design and evaluation of the maintenance process is vital to proper maintenance and continued safe/reliable operation. This VTB approach is unique in that is also allows design and evaluation of the maintenance aspect of the system design and it provides a "remote diagnostic server" (RDS) solution for serving "intelligent" IETMs to maintainers in a "thin client" manner. This environment can be used to help design the maintenance process, verify the procedures and diagnostic logic, and support maintenance training. The RDS environment centralizes the IETM and diagnostic logic for easy maintenance and upgrade. The TEAMATE architecture for this application will feature the complete separation of the core modules of TEAMATE, which include its kernel, from the TEAMATE implementation or application layers. Thus, the TEAMATE server will be cleanly separated into these two essential layers with a consistent and comprehensive Application Interface Layer (API) in between. The API layer in C/C++ will abstract all the lowest level services that TEAMATE offers to any third party application.

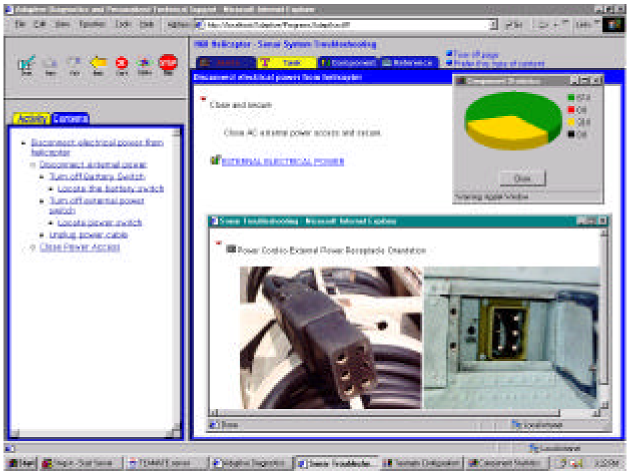

Disconnect electrical power from helicopter
Disconnect external power
Turn off Battery Switch
Locate the battery switch
Turn off external power switch
Locate power switch
Unplug power cable
Close Power Access
Close and secure
Close AC external power access and secure.
EXTERNAL ELECTRICAL POWER
Power Cord to External Power Receptacle Orientation

Figure 13 – Existing Web Based TEAMATE Example

AI-ESTATE compliance is a key piece in this architecture to ensure that the communication between IETM and TEAMATE is not only via open standards, network-aware protocols, but also the information exchange is also standardized. The architecture should be completely modular and provide the capability of exchanging the IETM or the diagnostic reasoner TEAMATE with other IETMs and diagnostic reasoners. The adoption and implementation of IEEE AI-ESTATE 1232.1-1997 and 1232.2 standards in TEAMATE allows the architecture to provide the above capability. The AI-ESTATE standard separates the knowledge captured in a system model from the reasoner and the test information. Consequently, any AI-ESTATE compliant reasoner can read any AI-ESTATE model. This provides the advantages of preserving the investment in capturing the knowledge about a system in models that are not tied to any reasoner and promotes reusability of model and test libraries.

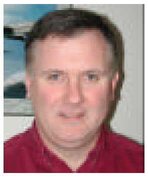
The 1232.1-1997 standard provides the formal specification of the data and knowledge necessary to perform system test and diagnosis. The specification of the diagnostic model allows the portability of the data and knowledge that is captured in the model. The 1232.2 standard provides the formal specification for the encapsulated services required of the diagnostic reasoner that directs the system test and diagnosis. Any AI-ESTATE compliant reasoner thus requires TEAMATE to be able to read 1232.1 compliant models and provide 1232.2 compliant services to be used by any IETM or other applications that can utilize those services. The AI-ESTATE 1232.1 standard has been in full-use and the 1232.2 standard will be a full use in the near future. Figure 13 provides a view of an existing web based TEAMATE/IETM solution to illustrate the capability that will be available to program engineers and maintenance personnel over the web at any location in the world.

7. SUMMARY

Reductions in funding and changing military requirements present a challenge to maintaining high levels of military readiness. It has become imperative that acquisition efforts yield cost-savings without reductions in performance. In particular, reductions in manning, such as the ones envisioned in the Navy's SC-21 and CVX programs, are required to yield significant reductions in labor expenses. However, reductions in personnel in complex human-hardware-software architectures will result in performance degradation, unless accompanied by significant changes in the overall system and a thorough analysis of the effects of various failure modes (e.g., design and human errors and hardware/software faults). In view of these new demands on design engineering and supportability, tools that can assist in evaluating potential failures, failure impact, diagnostics, safety, reliability, and maintenance are essential. Furthermore, these tools must be easy to use, graphical, and able to support all phases of the life cycle, a "cradle to grave" concept. The VTB approach described in this paper illustrates how a COTS Integrated Diagnostics tool set (the TEAMS^{TM} tool set) can provide a comprehensive design and support system for the entire life cycle of the system.

8. CONTRIBUTING PERSONNEL AND BIBLIOGRAPHY OF DIRECTLY RELATED WORK

QSI Author



Mr. Kevin Cavanaugh, *QSI's Chief Operating Officer, joined QSI in August 1999 after 18 years with Boeing. He has extensive project lead and management experience from projects such as NIMROD MRA4, F-22 Diagnostics and Health Management (DHM), JSF DHM, B-1B Expert Diagnostics System, and 757/767 Systems Engineering. With Boeing, Mr. Cavanaugh was an Associate Technical Fellow in the field of Integrated Diagnostics and Testability (Boeing's most senior engineer in the discipline). He most recently designed the Tactical Command and Sensor System (TCSS) diagnostics architecture for the NIMROD MRA4 upgrade. Mr. Cavanaugh was also a leader in the development of the F-22 DHM architecture and Boeing's JSF integrated diagnostics approach. Mr. Cavanaugh received a BS degree in Aeronautical Studies from Embry Riddle Aeronautical University in 1980.*

1.10 Bibliography

1. Sikorsky Internal Cost benefit study for the COSSI project, 1998.
2. An Onboard Real-time Aircraft Diagnosis and Prognosis System. Technical Progress Reports on NAS2-99048, July 26, 1999
3. "Joint Advanced Health and Usage Monitoring System - Advanced Concept Technology Demonstration", Phase I, Final Report, Sikorsky document no. SER 521365, August, 1998. Also, http://www.dt.navy.mil/jahums/ JAHUMS Project Homepage.
4. An Onboard Real-time Aircraft Diagnosis and Prognosis System. Technical Progress Report on NAS2-99048, October 26, 1999.
5. A Systematic Integrated Diagnostic Approach to Software Testing. Technical Progress Report on NAS2-99049, September 27, 1999.
6. S. Deb, S. Ghoshal, V. N. Malepati, and D. L. Kleinman, "Tele-diagnosis: Remote monitoring of large-scale systems", in Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 18-15, 2000
7. F. Wen, P.K. Willett, and S. Deb, "Signal Processing and Fault Detection with Application to CH-46 Helicopter Data", in Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 18-15, 2000
8. Somnath Deb, Amit Mathur, Peter K. Willett, and Krishna R. Pattipati, "De-centralized Real-time Monitoring and Diagnosis" in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 2998-3003, San Diego, October 11-14, 1998.
9. S. Deb, et. al. "QSI Integrated Diagnostics Toolset," 1997 IEEE AUTOTEST Conference, Anaheim, CA, September 1997.
10. S. Deb et. al. "Multi-Signal Flow Graphs: A novel Approach for System Testability Analysis and Fault Diagnosis," in Proc. IEEE AUTOTESTCON, Anaheim, CA, pp. 361-373, Sept. 1994.
11. S. Deb, et. al. "Multisignal Modeling for Diagnosis, FMECA, and Reliability" invited paper in 1998 IEEE SMC conference, San Diego, CA
12. S. Deb, et. al. "Multisignal Modeling for Diagnosis, FMECA, and Reliability" invited paper in 1998 IEEE SMC conference, San Diego, CA
13. B. Cameron, "Final Report on CH-46 Aft Transmission Seeded Fault Testing", Westland Research Paper RP907, Westland Helicopters, 1993.
14. P. Monson, M. Dwonczyk, and E. Manolakos, "Analog Neural Networks Based Helicopter Gearbox Health Monitoring System", Journal of the Acoustical Society of America, vol. 96(6), pp3235-3249, 1995.

15. G. Yen, “Health Monitoring of Vibration Signatures in Rotorcraft Wings”, Neural Processing Letters, vol.4(3), pp127-137, 1996.

16. Sudipto Ghoshal, Roshan Shrestha, Anindya Ghoshal, Venkatesh Malepati, Somnath Deb, Krishna Pattipati and David Kleinman, “An Integrated Process for System Maintenance, Fault Diagnosis and Support,” *Proceedings of the 1999 IEEE Aerospace Conference*, Aspen, Colorado, March 1999.