Multisignal Modeling for Diagnosis, FMECA, and Reliability

Somnath Deb, Sudipto Ghoshal, Amit Mathur, Roshan Shrestha and Krishna R. Pattipati
Qualtech Systems Inc.,
66 Davis Road, Storrs, CT-06268.
Ph/Fax: (860) 423-2099/8422,
E-mail: deb@teamqsi.com

ABSTRACT

Multisignal modeling methodology is a simple and efficient knowledge representation scheme that captures the basic attributes of a system (structure, specifications, etc.) that are obtainable from design data and product specifications. QSI's TEAMS toolset employs multisignal modeling for testability analysis, TPS development, on-board monitoring and ground support systems. In this paper, we outline the modeling methodology, its use in related areas of reliability and FMECA analysis, and our efforts in building a reusable test and model library.

INTRODUCTION

The original version of Qualtech Systems' Testability Engineering tool, TEAMS, 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, we introduced the multisignal modeling approach [1] 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 and multisignal modeling[1] beyond that of testability engineering. Our current research is focused on developing a complete solution package for Integrated Diagnostics (ID) that addresses all the facets of Design for Testability (DFT), Reliability Analysis, Failure Modes, Effects and Criticality Analysis (FMECA), and Test Program Set (TPS) development and field maintenance [2]. The use of the same model during DFT, Test Requirements Documentation (TRD), FMECA, TPS and PIMA development eliminates redundant modeling and ensures that the detection and isolation measures predicted in the design phase are realized in the field. Such an integrated approach could pay for itself within a year [3] via substantial reduction in lifecycle costs.

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

MULTISIGNAL MODELING

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, whereas 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.

Observations inspiring multisignal modeling

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

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

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).

Advanced multisignal modeling

Advanced features in the TEAMS toolset include:

- Signal grouping: 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.
- Signal aliasing: 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 [4] 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).

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.
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 SWITCHes.
   - 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 user 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.

A few applications of Multisignal Modeling

The following are a few examples of successful applications of the multisignal modeling methodology:

- The multisignal modeling methodology and real-time monitoring capability of TEAMS-RT were successfully employed on the integrated propulsion test bed of the X-33 reusable launch vehicle program by NASA-ARC [5] and Boeing NA (formerly Rockwell SSD). The models (two separate models for the liquid Hydrogen and Oxygen systems) had 44 operational modes and numerous signals spanning hundreds of components and tests!
- Using TEAMS, a 66% cost savings was achieved in the TPS development of a receiver-synthesizer board in the Joint Tactical Information Distribution System program. Instead of the usual test-centric approach prevalent in test community, the system information was captured in a multisignal model and then an optimized diagnostic strategy, complete with safety tests and mode changes, was generated by TEAMS. The TEAMS strategy achieved better diagnostic accuracy and fault isolation with fewer tests (211 versus 568) than the original manually-designed test program.
- QSI, working with Boeing, developed a TEAMS model of the 1553 data bus for flight control applications. As with other multisignal models, the 1553 model closely resembled the physical structure, and, hence, was easy to build and to validate. The full report and the model is available for download at http://www.teamqsi.com/downloads/pubs.html.
- Using TEAMS, Sikorsky Aircraft achieved a four-fold reduction in diagnostic costs for the anti-collision light control system of the Sea Hawk helicopter.
- Boeing Helicopters, Philadelphia, has built a model for the flight control system of the Comanche helicopter. The model, involving over 12,000 aspects and nearly as many tests, is the largest known dependency type model for any tool.

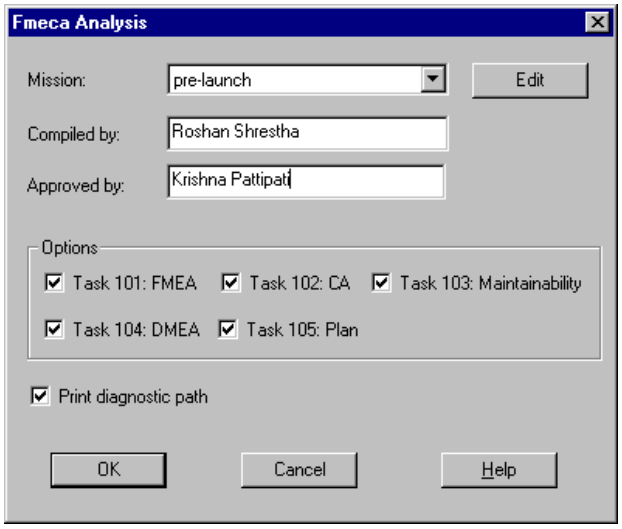

Figure 1: Screendump of FMECA Analysis options panel in TEAMS-5.0

- TEAMATE and HARVESTER are being implemented by Sikorsky for adaptive field maintenance with extensive multi-media documentation and repair tracking on the Black Hawk, Sea Hawk (SH60), and CH-53E helicopters. The diagnostic strategies for the blade-fold system of the Sea Hawk were successfully field-tested on a training simulator in April 1996. Significant reductions in testing time and cost were achieved.

FMECA 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
- 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 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.

Thus, the multisignal models capture sufficient information to substantially automate the Failure Modes, Effects and Criticality Analysis. TEAMS 5.0 is being updated to prompt the user for additional inputs necessary for FMECA (example: mission definitions, see Fig. 1) so that the information in the appropriate format (i.e., MIL STD 1629-A [6]) may be presented to the user.

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 l_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

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

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 TIME Mission 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 2: Sample reliability report for the receiver in 1553 bus system.

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

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

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 is presented in Figure 2.

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). We are actively developing such an RTML. At present, we have a prototype Reusable Test Library (RTL) as a part of the RTML. Boeing Defense and Space Group has also begun an internal effort to develop an RTL [8].

In our implementation of the RTML we treat every modular entity that comprises a mode as an independent object. According to the object-oriented paradigm, an object is a representation of a functional entity with defined interfaces [9]. The internal mechanisms of the functional entity are abstracted by the interfaces through which the external world manipulates the object. It is this fundamental property that promotes reusability of an object once its interfaces are known. Each entity of a model e.g., modules, tests, and switches that comprise a model, if defined appropriately, can be represented as an object and thus lend to their persistent storage and reusability. Currently, there are several efforts that seek to standardize the representation of objects and their interfaces that comprise the diagnostic models e.g., AI-ESTATE, ABBET, TERMS. These standardization efforts will further enhance the importance of the RTML approach in reducing the Total Cost of Model development as they will lead to a true plug and play capability of these objects even across application boundaries. 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

In our implementation, the RTML will store models and the individual entities that comprise the model like modules and test objects. It will also store fault lists and logistic data. We envision at least three views of any object: (1) TEAMS data structure; (2) Code in a high-level test programming language (e.g., C, CASS-ATLAS) for test objects; (3) Code in EXPRESS for data and knowledge representation of AI-ESTATE and TERMS-compliant models and test objects (4) Hyper-text Markup Language (HTML) and Standard

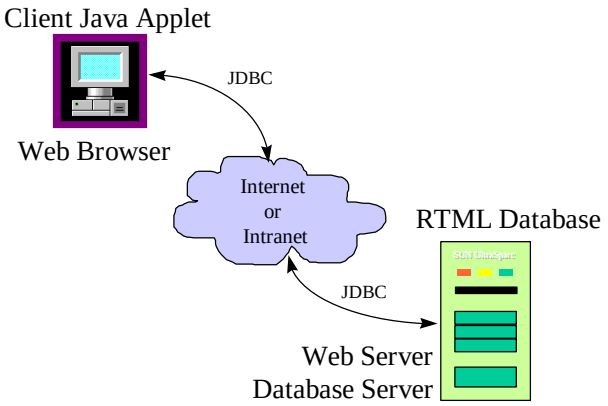

Figure 3 Schematic of the architecture of the web based Reusable Test Library.

Generalized Markup Language (SGML) documentation associated with models for easy integration with IETMs and other multimedia-rich diagnostic manuals. The first view is useful for (i) assessing UUT testability, (ii) generating an optimized diagnostic strategy, and (iii) generating code for a calling sequence. The second and third views are useful in automated test program development. The fourth view is useful in portable maintenance aid applications using TEAMATE a companion tool to TEAMS. At present, only the first and third views have been implemented. Seamless integration with our model development environment, TEAMS is also being developed.

One of the best features of the RTML is that it is an open client-server architecture where the client is an intuitively easy-to-use Java applet running in Netscape or Internet Explorer browser and the server is a database/Web server (see Fig. 3). Thus, by using the web based design and incorporating Java and JDBC, we achieve platform neutrality, database neutrality and location independence while providing security and access control - all of which should promote universal sharing of modeling information.

In the near future, our current prototype RTL will be expanded to store design and simulation model data, reliability data, logistic information and will be integrated with parameter estimation and data mining algorithms to create a comprehensive repository of information - TEAMS-KB.

CONCLUSIONS

Significant life cycle cost savings can be achieved via an integrated approach that addresses all the aspects spanning DFT, FMECA, reliability analysis, TPS development, on-board monitoring and field maintenance. This requires a common knowledge representation (i.e., multisignal models) that can be used at every stage of the system lifecycle, and a set of tools (i.e., TEAMS toolset [2]) that implements the pieces of the ID solution. This paper is a snapshot of our efforts to achieve such a comprehensive solution.

REFERENCES

1. 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. 1 1994.
2. S. Deb et al "QSI's Integrated Diagnostics Toolset", in *Proc. IEEE Autotestcon* 1997, Anaheim, CA.
3. Sikorsky Internal Cost benefit study for the COSSI project, 1998.
4. S. Deb, "Diagnostic Modeling of a Data Bus System and Software using TEAMS 4.0", final report submitted to Boeing Helicopters, PA.
5. A. Patterson-Hine *et al*, "Automated System Checkout to Support Predictive Maintenance for the Reusable Launch Vehicle" in *Proc. 1998 IEEE SMC conference*, San Diego, CA.
6. MIL-STD-1629-A Notice 2, 28 Nov. 1984, "Procedures for Performing a Failure Mode, Effects and Criticality Analysis", issued by CO, Naval Air Engineering Center, SESD, Code 531, Lakehurst, NJ 08733, Ph: 210 323 2326.
7. D.L. Iverson *et al*, "Digraph Reliability Model Processing Advances and Applications," Proceedings of the *AIAA Computing in Aerospace Conference*, 1993.
8. E. Cashar, "Development of a TPS reuse library using COTS tools", *AUTOTESTCON '96 Proceedings*, pp. 56-60, September, 1996.
9. G.Booch *Object-oriented design with applications*, The Benjamin Cummings Publishing, N.Y., 1991.