INTEGRATED DIAGNOSIS – A HIERARCHICAL APPROACH

John W. Sheppard
William R. Simpson
Advanced Research and Development Group
ARINC Research Corporation
2551 Riva Road
Annapolis, Maryland 21401

ABSTRACT

Increasing system complexity has led to major problems in system maintainability. With this increase in complexity, the difficulty of the diagnosis problem has risen dramatically. In the past, the development of test protocols to diagnose systems was spontaneous. Many system anomalies were diagnosed using the intuition and expertise of the system designer. As complexity increased, this approach gave way to more structured approaches. Small systems of intermediate complexity are currently being modeled and simulated at the gate level. However, larger and more complex systems either cannot be simulated or are too costly to simulate at the required level of detail. Therefore, a hierarchical approach to diagnosis of complex systems is required.

In the early 1970s, techniques for manufacturing, verifying, and testing were often combined, reinterpreted, and added to those tests developed to "bridge" the knowledge gap at the system diagnostic level. The results were less than spectacular. In the early 1980s, initiatives were undertaken to help keep pace with the growing complexity of maintenance. From these programs useful diagnostic products are now being developed.

In 1981, ARINC Research began to develop a hierarchical diagnostic modeling approach for system maintainability. This modeling approach has been applied throughout the various phases of system development life cycles. This modeling technique has had tremendous success, often yielding significant and spectacular improvements in system maintenance. Recent enhancements include the ability to develop portable maintenance aids and automatic test equipment (ATE) systems driven by the maintenance/diagnostic model. This paper reviews the basic modeling approach, the applications, and the hierarchy of analysis that can be completed using a single modeling architecture.

INTRODUCTION

Current system and test designs have resulted in 40% or higher false "pull" rates, the result of high ambiguity and labor-intensive test procedures. False alarms consume as much as 50% of maintenance resources. Studies of the CH-53[1] and F-16[2] had shown that troubleshooting actions can consume as much as 50% of the total labor hours spent for repair. Data for the scheduled airlines[3] show similar trends for complex electronics. These figures suggest a large potential return on investment if improved testability assessments and improved fault-isolation procedures are employed.

In the early 1980s, a number of industry and government initiatives were undertaken to help keep pace with the growing complexity of maintaining large systems. From these programs useful testing and diagnostic products are now being developed, and some are becoming household words in the automatic test community. Programs such as modular automatic test equipment (MATE),[4] intermediate forward test equipment (IFTE),[5] and integrated diagnostic support system (IDSS)[6] have spawned renewed interest in testability. The military has even developed a testability specification, MIL-STD-2165.[7]

Unfortunately, each of these initiatives either treated only one aspect of the life-cycle testability problem or treated each aspect of the life cycle as a separate issue. The integrated aspect of diagnostics has been the sharing of files and data and does not address philosophy and modeling approaches, except on a limited basis.

Since the mid-1980s, testability has been recognized as a valid and viable engineering discipline in areas beyond the board level and outside the manufacturing verification requirement. Equipment has good testability when existing faults can be

CH 2793-8/90/0000-0477 $1.00 © 1990 IEEE

confidently and efficiently identified. Confidence in testability systems is achieved when they frequently identify only the failed components or parts without requiring removal of good items. Efficiency is achieved by limiting the resources required (including laborpower, labor-hours, test equipment, and training).

Two approaches to integrated maintenance are described: current and hierarchical. The second approach is used by ARINC Research Corporation in its diagnostic aids System Testability and Maintenance Program (STAMP®) and Portable Interactive Troubleshooter (POINTER™). The information theoretic approach applied by ARINC Research to achieve hierarchical diagnosis is described in detail, as is the effectiveness of STAMP and POINTER.

INTEGRATED MAINTENANCE

CURRENT "INTEGRATED" APPROACHES

The current approach to integrating the maintenance process consists of combining multiple technologies through shared files and data. With this approach, uniformity breaks down because multiple representations of the maintenance problem and the system to be maintained are required, and complexity increases depending on the level of maintenance or the specific maintenance task. The different elements used to develop current integrated maintenance systems include a basic data base, analysis systems, built-in test (BIT), electronic simulation models, and expert system rule bases.

First, a "standard" maintenance data base is constructed and is the core of the current "integrated" approach. Analysis systems, BIT, electronic simulation models, and expert systems communicate with the data base, which contains representations of the system maintained that are appropriate for each element of the integrated package. The data base also contains other types of logistics data relevant to the maintenance problem.

A committee of "experts," associated with the different aspects of the maintenance problem, specifies the structure of this data base. This committee determines the overall maintenance architecture for the system, specifies the various modules of the integrated package and the functional packaging of the system for maintenance,* and specifies what parts of the system will be addressed by the various diagnostic modules.

At least three troubleshooting areas need to be addressed for most complex electronic systems: BIT, automatic test equipment (ATE), and manual troubleshooting (manuals or electronic aids). In order to determine what resources are required for each troubleshooting task, a testability assessment of the system is required. The resource allocation is frequently accomplished using undisciplined testability analyses or testability check lists (as prescribed in MIL-STD-2165). In the least effective cases, the committee compiles a wish list of troubleshooting capabilities without considering the testability of the system.

Once the testability resources have been allocated to the various troubleshooting tasks, the particular modules are constructed. First on-board diagnostics are addressed by specifying line-replaceable unit/line-replaceable module (LRU/LRM)-level BIT. This is most frequently achieved through improvised, trial-and-error specification of the tests. The result is a poorly defined system representation that is frequently inefficient and sometimes even wrong. The results of testing and corresponding maintenance actions are simply stored in the maintenance data base.

Once an LRU is pulled, further testing may be performed using automatic test equipment. Tests and test sequences for the ATE may be generated using circuit simulation at the gate level, and a test program set (TPS) test tape is generated for each unit under test (UUT). In such cases, the system representation consists of a simulation model and the set of tests to be run. Again, test results and maintenance actions are stored in the maintenance data base.

Finally, for the events when BIT and ATE are inappropriate or unavailable, manual troubleshooting procedures are defined. In order to continue integrating the complete maintenance process, the emphasis is placed on optimizing test procedures and making them available in electronic form. This is done by encoding maintenance manuals using an authoring system and gathering expertise to construct a maintenance expert system. This expert system then guides a maintenance technician through the troubleshooting process. Manuals, test results, and maintenance actions are stored in the central maintenance data base, and yet another system representation is required—the expert system rule base.

To summarize, current approaches to integrated maintenance consist of combining improvised BIT/BIT equipment, electronic simulation models, and expert system rule bases that were specified following an undisciplined testability assessment. Any "integration" comes solely from the maintenance data bases. The diagnostic approach or method of system representation remains unique to the particular level of diagnosis.

A HIERARCHICAL APPROACH

Another approach to integrated maintenance employs a single type of knowledge representation and applies a single approach to testability assessment and diagnosis. The knowledge base is analyzed for the testability assessment and guides fault isolation. The same form for representing a system can be used to determine BIT requirements, define TPSs for ATE, and guide the manual troubleshooting process.

A knowledge base for the hierarchical approach is the information flow model. The information flow model uses a data fusion approach to problem solving. In data fusion, a problem is solved by combining information from multiple sources to draw conclusions. In the case of troubleshooting, information gathered from performing a series of tests is combined to make a diagnosis. Defining the relationships between tests and conclusions to be drawn results in an information flow model, which is hierarchical by its very nature.

*As an alternative, the functional packaging may be left to the manufacturer.

The first step in the hierarchical approach to integrated maintenance is to develop a set of information flow models for the system to be maintained. Next, models are developed for onboard diagnosis (thus determining the requirements for BIT) and for each level of maintenance. Conclusions drawn at one level of isolation determine the appropriate model to use at the next level.

Once the models are developed, they can be analyzed to evaluate the testability of the system. Specification compliance can be verified, and design trade-offs can be performed in terms of improved testability. Thus, the modeling process can begin in early stages of system development. As the system progresses through its life cycle, the models are refined to reflect changes in the design.

For troubleshooting, the model defines available tests and inferences that can be drawn by using test outcomes. Thus the same models used to evaluate the testability of the system can be used directly for troubleshooting.

This hierarchical approach has been implemented in two software systems at ARINC Research. STAMP is used to develop information flow models, assess system testability, develop the diagnostic architecture, and define strategies for BIT, ATE, and manual troubleshooting in the form of fault trees. POINTER serves as an intelligent controller for BIT, ATE, and manual troubleshooting if more flexibility is required than that provided by STAMP-generated fault trees. POINTER extends the hierarchical approach to integrated maintenance in that it uses the STAMP information flow models directly. The following sections further describe the STAMP/POINTER approach.

DIAGNOSTIC STRATEGY

A fault-isolation strategy is a road map showing how to use the available tests to determine what in the system, if anything, has failed. Three of a number of different strategies are discussed below: sequential, half-interval, and adaptive.

SEQUENTIAL FAULT ISOLATION

Sequential fault isolation, the most common approach, proceeds along the functional flow of the system. In the example of Figure 1, an evaluation of test 5 is made first. If test 5 is *good*, we are finished, and the result is a Retest Okay (RTOK). If the test is *bad*, we proceed to test 4 (to test the preceding component in the functional flow). If that test is *good*, we are finished, and we know that component 5 is bad. If the test is *bad*, we proceed to test 3, and so on. This sequential or signal tracing approach is called a directed search. It is the method most technicians use in the absence of detailed procedures. For a serial system, such as our example, if all events* are equally likely to occur, it will take an average of $(n - 1)/2 + (n - 1)/n$ tests to isolate one of the events, where n = the number of conclusions that can be drawn. For the example, it takes an average of 3.86 tests with the least number of tests being 1 (RTOK) and the largest number of tests being 6 (input and component 1).

HALF-INTERVAL TECHNIQUE

An alternative to this strategy would be to use system partitioning. When using this strategy, any given test, depending on its results, eliminates certain events from being the failure cause. Each test partitions the possible results into two states, feasible and infeasible. In the example, if test 3 is *bad*, under a single-failure assumption, the failure could not have been caused by component 4 or component 5. Those two results go into the infeasible category. RTOK is also infeasible, so it goes into the infeasible category. Under these criteria, fault isolation is achieved when only one result remains feasible. We can expect to put the largest number of results in an infeasible state, regardless of outcome, if we choose a test near the middle of the system. This would be either test 2 or test 3 for our example, depending on how one rounds off to get the middle or halfway point. This method of partitioning to the middle or halfway point is termed half-interval.

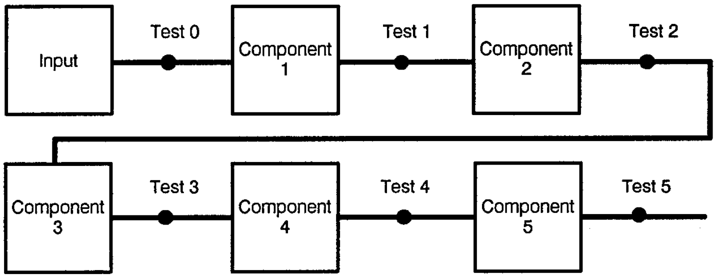

FIGURE 1. SERIAL SYSTEM FOR EXAMPLE PURPOSES

*An event is the failure of a component, the failure of an input, or RTOK. A total of n events is possible. For Figure 1, n = 7 (5 components, 1 input, and RTOK).

Fault isolation can be mathematically described as a partition process. Let $\mathbf{C} = (c_1, c_2, \ldots, c_n)$ represent the set of components. After the j^{th} test, a fault-isolation strategy partitions **C** into two classes.

$\mathbf{F}^j = (c_1^j, c_2^j, \ldots, c_m^j)$ = the set of components that are still failure candidates after the j^{th} test (feasible set).

$\mathbf{G}^j = \mathbf{C} - \mathbf{F}^j$ = the set of components found to be good after the j^{th} test (infeasible set).

By this structure, a strategy will have isolated to the failure when $\mathbf{F}^j$ consists of a single element or a component ambiguity group. From the earlier discussion of serial systems, it is seen that the directed search strategy may reduce only the size of $\mathbf{F}^j$ by one component at a time. The half-interval technique reduces $\mathbf{F}^j$ by approximately 50% after each test, an obvious advantage.

It can be proved that for a well-ordered system, the half-interval technique will provide the minimum number of tests. However, such an ordering rarely exists. The STAMP approach uses an adaptive, information-based strategy, which is discussed in the next section.

AN ADAPTIVE STRATEGY

Test results impart information. The type, amount, and quality of such information should be considered when developing a fault-isolation strategy. For our purposes, we assume equal quality in the test results in that a *good* or *bad* indication of a test actually reflects the state of the UUT. However, this assumption may be relaxed.[8]

The amount of information provided by different tests is quite variable. Referring to Figure 1, if the first test reading was at test 5 and it was bad, the only inference we can make is that one or more of the six elements is bad and RTOK is not possible. On the other hand, a bad reading at test 0 specifically tells us that the input is bad. However, we cannot conclude that test 0 is a better test to start with. For example, take the case of a good reading: good reading of test 5 tells us that all elements are good (RTOK), while test 0 good tells us only that the input is good. This type of information distribution leads to the basic premise that a good test at the end of the functional flow is information-rich, as is a bad test early in the functional flow. If we can hypothesize a linear variation in information content, we have a relationship similar to that shown in Figure 2.

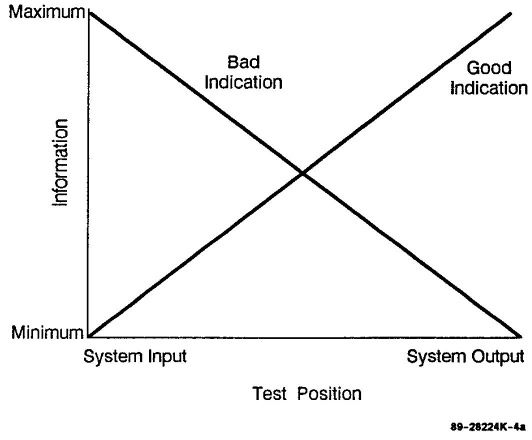

FIGURE 2. LINEAR INFORMATION DISTRIBUTION

The fault-isolation process involves considering the tests as having an unknown outcome; therefore, a reasonable strategy for test choice is to balance the information content. Indeed, that is precisely the basis for the theoretically optimum half-interval strategy. Unfortunately, a linear information assumption may not be appropriate for complex designs.

In seeking to overcome the limitations of the half-interval technique, it is apparent that if all dependencies in a system are known, the information content of each test can be calculated. If a test is performed, knowing the set of dependencies allows us to draw conclusions about a subset of components. The process of drawing conclusions about the system from limited information is called inference.

For any test sequence, STAMP and POINTER allow us to compute $\mathbf{F}^j = (c_1^j, c_2^j, \ldots, c_k^j)$ and the set of remaining failure candidates, namely $\mathbf{F}^{j1}, \mathbf{F}^{j2}, \ldots, \mathbf{F}^{jk}$. An algorithm has been developed to look at the information content of all remaining tests so that the number of remaining tests that have to be performed to isolate faults is minimized over the set of potential failure candidates. This adaptive approach embodies several artificial intelligence algorithms, including inference and pattern recognition. It can be described mathematically as follows:

- Let **D** represent the full dependency relationship between components and test points. (This is formulated as a matrix representation.)
- Let $\mathbf{S}_k$ be a sequence of k tests, $(t_{j1}, t_{j2}, \ldots, t_{jk})$.
- Let $\mathbf{F}^k$ be the feasible failure candidate set associated with $\mathbf{S}_k$.

We then develop an information measure for each remaining (unperformed) test (j), which is a function of the dependency relationship and the remaining candidate failure class, say, $\mathbf{I}_k^j = f(\mathbf{D}, \mathbf{F}^k)$. The test sequence $\mathbf{S}_k$ that is derived is obtained by optimizing at each decision point. That is, the next test in the sequence is taken as the test that maximizes $\mathbf{I}_k^j$ for the conditions imposed by each previous test outcome and is based on an unknown current outcome. The sequence ends when adequate information is derived for fault isolation.

To this point, we have considered only the test point location and functional or signal flow in discussing isolation strategies. Underlying that discussion is the assumption that all failures are equally likely and that all tests require equal resources. In practice, such an assumption may be unacceptable. Ideally, a fault-isolation strategy should give more weight to tests that can determine the status of components most likely to fail and tests that are simple to perform or easily assessable.

STAMP and POINTER allow this type of information to be easily incorporated into the information measure. They also allow for data on component failure rates, test times, and costs to be incorporated directly into the search strategy algorithm. The logistic or maintenance manager can then select a fault-isolation strategy that minimizes resources by selecting one or more of the weighting factors.

POINTER can adapt its troubleshooting process to changing diagnostic conditions through a process of learning. During a fault-isolation session, POINTER times the tests as they are performed. The test times are then recorded and combined with previously recorded test times to derive a new test time measure. This measure is used by POINTER to improve fault-isolation performance when attempting to minimize the time required to fault-isolate.

In addition, POINTER records the repairs made to the system with the current number of hours of system operation. Failure rates are then recomputed on the basis of the repairs and operating hours, and the new failure rates are used by POINTER to select appropriate tests when attempting to isolate failures by weighting failure probability.

In addition to recording test times and failure rates for improving fault-isolation performance, POINTER maintains two sets of files that can be used in logistics documentation. First, the learning file associated with each POINTER model contains information on test times, skill level, failure rates, number of recorded failures, the most recent operating hours for each repair, and a link to a log file. The second set of files comprises log files.

Each fault-isolation session creates a log file containing information about that session. Each log file includes a description of the setup conditions for fault isolation, a record of the test sequence, a list of failures identified, any repair actions, and comments provided by the technician. Further, the test sequence information includes the times to perform each test, all POINTER actions taken by the technician, and the test outcome. If learning takes place, information on how test times and failure rates changed is also included. Finally, each log file is linked to the previous log file associated with a repair of the same failure (if one exists).

The information provided in these files does not include the results of any logistics analyses. It does, however, provide some of the data required for such analyses, or other files can be used with a separate documentation system that records and analyzes logistics information.

EFFECTIVENESS OF THE HIERARCHICAL APPROACH

The ability to improve fielded system testability and maintainability by 50% to 100% ranks our hierarchical approach (STAMP) as one of the most successful ARINC Research technological applications. The model-based analysis technique has been applied to more than 50 complex systems covering almost every engineering discipline. Table 1 lists some of those applications. The values in Table 1 resulted from field-monitored applications, prototype testing, and side-by-side testing by expert maintenance technicians.

As an example of a model-based diagnostic application, ARINC Research, under contract to the Naval Sea Systems Command, Shipyard Training Division (SEA 072), was tasked to develop a prototype computer-based training aid for diagnosis and fault isolation of a typical shipboard electronic system.[8] This system was to provide a bridge between classroom training and on-the-job training and to serve as an efficient maintenance aid for shipyard technicians. The Mk 84 Static Frequency Converter, installed on AEGIS-class cruisers and destroyers, was selected for the application from among several alternatives. Analysis revealed that a significant savings in fault-isolation time could be achieved by applying model-based diagnostics rather than the currently documented procedures. By the end of the project, the number of steps required for a system checkout was reduced by 71%. The significant results are listed in Table 2.

For the Mk 84 Static Frequency Converter, a static fault tree was developed for implementation on the portable maintenance aid. (Note that we refer to this type of maintenance aid as an "electronic manual" or "electronic fault tree"; it is not an intelligent aid such as POINTER.) The computer chosen for the portable maintenance aid was the GRID Systems Corporation portable computer. The GRID computer is a 512-kilobyte MS–DOS compatible system with a battery power pack. It is about the size of a portable typewriter (approximately 12 by 16 by 2 inches) and weighs approximately 11 pounds.

Statically generated diagnostic strategies were implemented on the GRID in an interactive question-and-answer format that provided detailed test procedures and repair procedures. The system was field-verified at Dahlgren, Virginia, by a group of Mk 84 maintenance technicians who used either the maintenance aid, technical manual procedures, or their own expertise. The control of the experiment was by fault insertion. Maintenance technicians not using the GRID were allowed to use anything normally at their disposal, including the technical

TABLE 1. RESULTS OF MODEL-BASED DIAGNOSTIC APPLICATIONS

System	Results
AN/ALR-62	Reduced ambiguity groups by more than 40%.
IFC Air Handling System	Unique isolation improved by more than 100%.
AN/MSQ-103C TEAMPACK Track EW Vehicle	Reduced required testing by 87%. Portable maintenance aid developed.
Mk 84 60/400 Hz Static Frequency Converter	Reduced required testing by 70%. Portable maintenance aid developed.
UH-60A Stability Augmentation System	Reduced mean time to fault-isolate by factor of 10; reduced maintenance complexity by factor of 3.
ALQ-131 EW Pod System	Reduced mean time to fault-isolate by 75%.
ALQ-184 EW Pod System	Reduced false alarm rate by factor of 10. Developed UUT software procedures.
B-2 Bomber DFT Program	Improved specification compliance at the shop-replaceable unit (SRU) level by 80%.
Tokyo Electric Power Company 11-MW Fuel Cell	Developed POINTER portable maintenance aid for site use.
GUARDRAIL Relay System	Developed POINTER portable maintenance aid for field use.
AF8B Power Supplies	Developed POINTER portable maintenance aid for shop use.

TABLE 2. FAULT-ISOLATION COMPARISON OF Mk 84 TECHNICAL MANUAL WITH MODEL-BASED STATIC FAULT ISOLATION

Fault	Technical Manual (Steps)	STAMP (Steps)	Reduction (%)
Total System Checkout in Rectifier Unit	320	90	71
Circuit Breaker Control Signal	26	10	61
System Stop Signal	25	6	76
Rectifier Protection Board Output	Min 28 Max 48	Min 5 Max 25	83
System Control and Protection Board Output	Min 30 Max 75	Min 8 Max 20	73 73
Fuse Open Signal	27	3	89

manuals, intuition, and past experience. The results of the field verification were as follows:

- On each fault insertion, the maintenance technician with or without the use of the GRID performed better than with the procedures recommended by the existing technical manual.
- For each fault insertion the performance results obtained by the maintenance technician who used the GRID were equal to or better than the performance results achieved by an experienced maintenance technician who did not use the GRID.

SUMMARY

The hierarchical model-based approach to integrated maintenance differs from other approaches in several significant ways. Most important, the model-based approach provides a truly integrated approach to diagnosis. STAMP and POINTER use this approach, thus taking advantage of the following attributes:

- The single form of knowledge representation enables all diagnostic elements to function in a consistent manner, regardless of the type or level of maintenance.
- This knowledge representation can be used for testability analysis, including maintenance architecture and functional packaging.
- The models are hierarchical, making them easily adaptable to all levels of maintenance.
- Because information theory and data fusion define the framework of developing the models, the approach may apply to many engineering disciplines, including hybrids.
- The approach permits diagnosis to be dynamically tailored to the current context (i.e., known information and available resources).
- The models facilitate effective testability assessment, intelligent troubleshooting, and direct links to logistics data bases.

Thus STAMP and POINTER, by using the information flow model, permit all aspects of the maintenance process to be addressed using a single method of knowledge representation and a single method of knowledge base processing.

REFERENCES

1. Thomas N. Cook, et al., "Analysis of Fault Isolation Criteria/Techniques," Proceedings of the Annual Reliability and Maintainability Symposium, San Francisco, California, January 1980.
2. M. L. Labit, et al., *Special Report on Operational Suitability (OS) Verification Study Focus on Maintainability*, Publication 1751-01-02-2395, ARINC Research Corporation, Annapolis, Maryland, February 1981.
3. Aeronautical Radio, Inc., Avionics Maintenance Conference Report—San Diego, 1987, Publication 87-087/MOF-34, Annapolis, Maryland, August 1987.
4. G. Cross, "Third Generation MATE—Today's Solution," Proceedings of the 1987 IEEE AUTOTESTCON Conference, San Francisco, California, November 1987.
5. C. M. Espesito, et al., "U.S. Army/IFTE Technical and Management Overview," Proceedings of the 1986 IEEE AUTOTESTCON Conference, San Antonio, Texas, September 1986.
6. J. R. Franco, "Experiences Gained Using the Navy's IDSS Weapon System Testability Analyzer," Proceedings of the 1988 IEEE AUTOTESTCON Conference, Minneapolis, Minnesota, September 1988.
7. Naval Electronic Systems Command (ELEX-8111), *Testability Program for Electronic Systems and Equipment*, MIL-STD-2165, Washington, D.C.: Naval Electronic Systems Command, January 26, 1985.
8. W. R. Simpson and J. W. Sheppard, "The Application of Evidential Reasoning in a Portable Maintenance Aid," AUTOTESCON '90, San Antonio, Texas, September 1990.
9. P. Gregory, "Test Evaluation for a Prototype Computer-Based Training Aid," Publication 3467-01-01-4321, ARINC Research Corporation, Annapolis, Maryland, March 1987.