N94-11563

DESIGN FOR TESTABILITY AND DIAGNOSIS AT THE SYSTEM-LEVEL

by

William R. Simpson
John W. Sheppard

ARINC Research Corporation
2551 Riva Road
Annapolis, MD 21401

ABSTRACT

The growing complexity of full-scale systems has surpassed the capabilities of most simulation software to provide detailed models or gate-level failure analyses. The process of system-level diagnosis approaches the fault-isolation problem in a manner that differs significantly from the traditional and exhaustive failure mode search. System-level diagnosis is based on a functional representation of the system. For example, one can exercise one portion of a radar algorithm (the Fast Fourier Transform [FFT] function) by injecting several standard input patterns and comparing the results to standardized output results. An anomalous output would point to one of several items (including the FFT circuit) without specifying the gate or failure mode. For system-level repair, identifying an anomalous chip is sufficient.

We describe here an information theoretic and dependency modeling approach that discards much of the detailed physical knowledge about the system and analyzes its information flow and functional interrelationships. The approach relies on group and flow associations and, as such, is hierarchical. Its hierarchical nature allows the approach to be applicable to any level of complexity and to any repair level. This approach has been incorporated in a product called STAMP® (System Testability and Maintenance Program) which has been developed and refined through more than 10 years of field-level applications to complex system diagnosis. The results have been outstanding, even spectacular in some cases. In this paper we describe system-level testability, system-level diagnoses, and the STAMP analysis approach, as well as a few STAMP applications.

INTRODUCTION

System-level diagnosis has always been an afterthought in system design. Initially (i.e, circa 1930) system-level failures announced themselves. Parts fell off, items quit working, or the failure symptom itself pointed to the subsystem that demanded repair. As systems became more complex a symptom indicated that a failure was restricted to a small list of possible causes. Further testing was undertaken to localize the failure to a level consistent with repair.

As systems have grown in complexity we have been forced to rely on testing that is an outgrowth of product assurance rather than on field-derived maintenance information. The easiest obtainable test information has been that developed from testing by the manufacturer during equipment production. At the same time, the product assurance people placed their resources on intermediate production screening. Realizing that system-level diagnosis was an extremely complex problem, the manufacturer began to screen incoming parts and to test at the detailed subassembly level in an effort to avoid delivering a malfunctioning system. What resulted was a mismatch; that is, the tests that were available to the field technician were not developed for system-level diagnosis, but rather, for system verification purposes. In fact, the tests were designed to avoid any situation where system-level diagnosis was required.

Because of this mismatch, system and test design provided diagnosis that frequently resulted in 40% or higher false "pull" rates, the result of high ambiguity and labor-intensive test procedures, and

false alarms consumed excessive maintenance resources. Studies of the CH-54 and the F-16 showed that troubleshooting actions consumed as much as 50% of the total labor-hours spent for repair.[1] Data for the scheduled airlines revealed similar trends for complex electronics.[2] When systems were sent back to the factory, a bench check was performed and only two outcomes resulted:

- A retest OK indicating improper diagnosis in the field or inadequate bench checking
- An anomalous system to be discarded or dissected for subassembly test

Both of these outcomes are unacceptable.

The situation in system diagnosis continued to deteriorate, and the need for system-level diagnosis was easily recognizable in the late 1970s and early 1980s. Readiness levels for military aircraft were often low, with as few as 50% of the assets available in some maintenance cycle. In the early 1980s, several initiatives such as MATE, IFTE, and CASS were underway, and a number of tools were being developed, such as IDSS, STAMP and I-CAT.[3-8] All of these, to one extent or another, addressed the system-level aspects of testability and diagnosis. The first military specification for testability (MIL-STD-2165) became effective in 1985.[9]

SYSTEM LEVEL TESTABILITY

According to MIL-STD-2165, testability is defined as:

A design characteristic which allows the status (operable, inoperable, or degraded) of an item to be determined and the isolation of faults within the item to be performed in a timely and efficient manner.[9]

The literature generally discusses different types of testability when referring to system-level testability: *inherent* and *achieved* testability. Inherent testability addresses the way the system is designed and encompasses the ability to observe system behavior under a variety of stimuli. Inherent testability is defined by the location, accessibility, and sophistication of tests and test points that may be included in the system. Achieved testability addresses how the system is maintained. It is defined by the results of the maintainability process (such as false alarms, ambiguities, incorrect isolations, no faults found). Note that the *achieved* testability has the *inherent* testability as a goal and no testability as a lower limit.

During the design phase, the testability analysis should provide the following information related to the inherent testability of the system:

- ***Ambiguity Groups***—Components which are and components which are not uniquely identifiable in the current system/test configuration.
- ***False Failures***—When multiple failures occur, any combinations that can provide the same symptoms as an unrelated single failure.
- ***Hidden Failures***—When multiple failures occur, their relationship, if any, and the root cause of the failure hidden.
- ***Information Feedbacks***—Cycles of diagnostic information. Feedbacks typically cause isolation problems and result in larger-than-acceptable ambiguity groups. Mapping feedback is one of the first steps in improving testability by reducing ambiguity groups.
- ***Nondetections***—Components that have failure modes which are not observed by any of the available tests.
- ***Test Disposition***—Necessary additional and unnecessary tests. Eliminating unnecessary tests reduces maintenance complexity and test program set (TPS) test times.
- ***Tolerance to False Alarms***—Any special provisions required by the system to handle potential false alarms.
- ***Operational Isolation***—The probability that one can expect to isolate 1, 2,... or fewer replaceable units. This information is critical for logistic planning.

DIAGNOSIS AT THE SYSTEM LEVEL

As with testability, diagnosis often refers to more than one concept. In this paper, three basic terms are used with the diagnosis descriptions: *detection*,

localization, and *isolation*. Detection refers to the ability of a test, combination of tests, or a diagnostic strategy to identify that a failure in some system element has occurred. This term is often associated with built-in test (BIT) and may actually be the design criterion set upon BIT.

Localization refers to the ability to restrict a fault to some subset of possible causes. This also is associated with a combination of tests or a diagnostic strategy. Clearly, all BIT that can detect must also localize (to at least one of all possible faults). If the localization is sufficient in most cases to undertake repair, we often refer to the BIT as *smart* BIT. BIT, however, is not the only diagnostic technique that localizes. Often automatic test equipment (ATE) and manual isolation techniques use a diagnostic strategy that localizes the fault to a degree sufficient to undertake repairs.

Isolation is often misused to represent that localization has been achieved to a degree consistent with a single repair unit. Actually, it means that, through some test, combination of tests, or diagnostic strategy, the specific cause of a fault has been identified.

A diagnostic strategy should provide a limited set of items:

- A procedure that brings the achieved testability up to the level of the inherent testability.
- A procedure that can fault-isolate (localize) the system while optimizing one or more criteria.

THE STAMP APPROACH

It is assumed that, at any analysis level, when an engineer writes a full-scale physical simulation of the entire system at a specific level of detail, he or she will then be able to answer all of the testability questions by meticulously tracing stimuli through the system to observe responses. This is possible when faults are exhaustively modeled, and the engineer can determine such items as nondetection and ambiguity. Unfortunately, because of the sheer volume of computations required at higher levels of complexity or by a larger system, this is not practical. For example, suppose that we have a very large-scale integrated (VLSI) chip with 10,000 gates, any one of which may be "stuck open" or "stuck at," yielding 20,000 faults to model. If 4 such chips are on board with other components, and 6 such boards make up the digitizer in a color radar display that has 23 such subsystems, we have to model at least 11 million failure modes!

When we began to develop a less computationally intense process, we wanted to build an analysis method that is hierarchical and discards a fair amount of the detail carried along in a physical representation. First, we strip the test of its stimulus-response details and turn it into an information carrier. This is not to say that the details of how the test is conducted are unimportant. In fact, they are essential in actually performing the test. We simply do not carry them along in our analysis (but we do pick them up later). Second, we ignore the details of gates, resistors, and hardware implementations and, rather, consider functions. The latter gives us a hierarchical formulation because functions can be aggregated from combinations of other functions, and we can proceed functionally to any level in the analysis. (A function, of course, carries with it an aggregation of hardware or a piece of hardware.) This in turn provides a way to "repair" functions.

What have we lost? A great deal. We can no longer use our model to provide the stimulus-response details. A computer-aided drawing (CAD) file can no longer be used directly for input, although we may be able to enter some of the details through translation. The solution may be a much grosser localization than a simulation model.

What have we gained? A great deal. We can now perform our testability analysis in a hierarchical manner. We can hypothesize information sources without concerning ourselves with the details of stimulus-response—until and unless we want to actually perform the test. We can play what-if and conduct trade-off analyses at a much simpler modeling level. And we have a full range of information theoretic tools to help us answer the basic testability and fault-isolation questions.

One tool, STAMP, derives measures of testability and synthesizes fault-isolation strategies on the basis of an information flow model of the system under analysis. It is important to understand the fundamentals of information flow modeling and fault-isolation theory. The vehicle for information

flow modeling is a block diagram that represents the functional topology of a given system. Additional data available for the model include hierarchical grouping, special inference, and cost and other weighting criteria. A full range of testability measures and tables is then produced to provide the basic information listed in the "Testability at the System Level" section. The specifics as they apply to the STAMP analysis are detailed in references 10 and 11, which include example computations.

Fault isolation can be described mathematically as a partition process. Let $C = (c_1, c_2,...,c_n)$ represent the set of n components. After the jth test, a fault-isolation strategy partitions C into one of two classes:

$$F^j = (c_1^j, c_2^j,..., c_m^j) \qquad (1)$$

where F^j is the set of components that are still failure candidates after the jth test (feasible set), and m is the number of components in the set. The complement of this set is given by:

$$G^j = C - F^j \qquad (2)$$

where G^j is the set of components found to be good after the jth test (infeasible set). This set will contain m-n components.

By this structure, a strategy will have isolated a fault when F^j consists of a single element or can no longer be subdivided (F^j consists of a component ambiguity group).

It can be proved that for a well-ordered system, a half-interval search technique will provide the minimum number of tests; however, such an ordering rarely exists. The STAMP approach uses an adaptive, information-based strategy, because in seeking to overcome the difficulty of ordering a system for the half-interval technique, it became apparent that if all dependencies in a system were known, the information content of each test could be calculated. If a test is performed, 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 allows us to compute $(c_1^j, c_2^j,..., c_k^j)$ and the set of remaining failure candidates, namely F^1, F^2,..., F^j. An algorithm has been developed to look at the information content of all remaining tests so that the number of remaining tests that must be performed to isolate faults is minimized over the set of potential failure candidates.

STAMP EFFECTIVENESS

It can be shown that for a well-ordered or straightforward serial design, STAMP reduces to the half-interval technique, which is known to be optimal for that case. Unfortunately, the general case is known to be NP-complete,[12] so we are forced to rely on an approximate solution. In a number of applications, the adaptive, information-theoretic approach has provided the mean and the variance of the required number of tests under all failure conditions, either equal to or lower than those resulting from other procedures examined, and often approaching the theoretical minimum values. Table 1 lists a few of the more than 250 systems analyzed by STAMP.

SUMMARY

STAMP emphasizes diagnosis at the system level. This emphasis differs from most other testability analysis tools that operate at the gate or, at most, board level. This system-level emphasis enables STAMP to be hierarchically applied and enables the engineer to approach the testability problem from an information flow standpoint rather than from an electronic simulation. An additional result is that the approach is independent of the underlying technology, thus allowing the analysis of most systems, including hybrids. A shortcoming of this approach is that it cannot be used to directly develop the detailed definitions of the tests. STAMP has been applied to many types of systems, and these applications have been for a large number of system technologies and at varying points in the system life cycle. The results indicate that there is a large potential gain in providing system-level testability and diagnosis analyses.

Table 1. Results of STAMP Applications

System	Customer	Results
ALR-67 Countermeasures Set	NADC/USN	Developed test procedures for TRDs
ALR-62 Warning Receiver	ALC/USAF	Reduced ambiguity groups by over 40%
Air Pressurization System	Int'l Fuel Cells	Unique isolation improved by over 100%
MSQ-103C TEAMPACK	EW/RSTA/USA	Reduced required testing by 87%; portable maintenance aid developed
Mk 84 60/400 Hz Static	NAVSEA/USN	Reduced required testing by 70%; portable frequency converter maintenance aid developed
UH-60A (Black Hawk) Stability Augmentation System	ATL/USA	Reduced mean time to fault-isolate by factor of 10; reduced maintenance complexity by factor of 3
ALQ-131 Podded EW System	ASD/USAF	Reduced mean time to fault-isolate by 75%
ALQ-184 Podded EW System	AFLC/USAF	Reduced false-alarm rate by a factor of 10; developed UUT software procedures
B-2 Bomber DFT Program	USAF/Northrop	Improved specification compliance at the shop replaceable unit (SRU) level by 80%

REFERENCES

1. Cook, Thomas N., et al. "Analysis of Fault Isolation Criteria/Techniques," *Proceedings — Annual Reliability and Maintainability Symposium*, San Francisco, CA, January 1980.

2. Aeronautical Radio, Inc. *Avionics Maintenance Conference Report — San Diego*, 1987. Publication 87-087/MOF-34, Annapolis, MD, August 1987.

3. Cross, G. "Third Generation MATE—Today's Solution." *Proceedings of the 1987 IEEE AUTOTESTCON Conference*, San Francisco, CA, November 1987.

4. Espesito, C. M., et al. "U.S. Army/IFTE Technical and Management Overview." *Proceedings of the 1986 IEEE AUTOTESTCON Conference*, San Antonio, TX, September 1986.

5. Najaran, Captain M. T., "CASS Revisited—A Case for Supportability," *Proceedings of the 1986 IEEE AUTOTESTCON Conference*, San Antonio, TX, September 1986.

6. Franco, J. R. "Experiences Gained Using the Navy's IDSS Weapon System Testability Analyzer," *Proceedings of the 1988 IEEE AUTOTESTCON Conference*, Minneapolis, MN, September 1988.

7. Simpson, W. R., and Sheppard, J. W., "Experiences with a Model-Based Approach to the Fault Detection and Isolation of Complex Systems," *Symposium on Artificial Intelligence Applications in Military Logistics*, Williamsburg, VA, March 1990.

8. Cantone, R. A., and Caserta, P., "Evaluating the Economical Impact of Expert Fault Diagnosis Systems: The I-CAT Experience," *3rd IEEE International Symposium on Intelligent Control*, Arlington, VA, August 1988.

9. *Testability Program for Electronic Systems and Equipment*, MIL-STD-2165, Washington, DC, Naval Electronic Systems Command, January 1985.

10. Sheppard, John W., and Simpson, William R., "A Mathematical Model for Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 8, No. 4, December 1991, pp. 25-38.

11. Simpson, William R., and Sheppard, John W., "System Testability Assessment for Integrated Diagnostics," *IEEE Design and Test of Computers*, Vol. 9, No. 1, March 1992, pp. 40-54.

12. Hyafil, Laurent, and Rivest, Ronald L., "Constructing Optimal Binary Decision Trees is NP-Complete," *Information Processing Letters*, Vol. 5, No. 1, May 1976, pp. 15-17.

Biographical Sketches of the Authors

John W. Sheppard holds a B.S. in Computer Science from Southern Methodist University and an M.S. in Computer Science (emphasizing Artificial Intelligence) from The Johns Hopkins University, where he is currently a Ph.D. candidate in Computer Science. His research areas include fault diagnosis, machine learning, neural networks, and knowledge representation. His work has included the development of A.I. techniques and algorithms that are being applied in system diagnosis, knowledge base verification, and software classification. Mr. Sheppard is one of the principal developers of an intelligent, interactive maintenance assistant (Portable Interactive Troubleshooter -- POINTER) which guides a diagnostic process through test choice and evaluation, explains reasoning, and incorporates elements of evidential reasoning and neural networks to process information obtained from uncertain or incomplete testing. Mr. Sheppard is a senior research analyst in the Advanced Research and Development Group at The ARINC Research Corporation. He can be reached by e-mail at sheppard@cs.jhu.edu

William R. Simpson holds a B.S. in Aerospace Engineering from the Va. Polytechnic Institute and State University, an M.S.A. in Engineering Management from the George Washington University, as well as, an M.S. and a Ph.D. in Aerospace Engineering from the Ohio State University. He is also a graduate of the U.S. Naval Test Pilot School in Patuxent River, Maryland. His work in the area of testability and fault diagnosis resulted in the development of the System Testability and Maintenance Program (STAMP) which uses an information flow model to assess system testability and generate efficient fault isolation strategies. He was also a principal developer of the POINTER system which includes reasoning under uncertainty, similarity and explanation based learning, logical inference, and decision optimization. In addition to STAMP and POINTER, Dr. Simpson applied the information modeling approach to non-cooperative target recognition (Non-Cooperative All-Source Target Identification -- NASTI) and to electronic warfare signal sorting (Signal Evaluation for Emitter Recognition -- SEER). He has also participated in the development of neural networks for software classification and reasoning termination. Dr. Simpson is a research fellow in the Advanced Research and Development Group at the ARINC Research Corporation. He can be reached by e-mail at wsimpson@mcimail.com