INTEGRATING TESTABILITY ANALYSIS TOOLS WITH AUTOMATIC TEST SYSTEMS (ATS)

Joseph Pillari, Thaddeus (Ted) Pertowski, and Adelbert Protin
GEC Marconi
Electronic Systems Corporation
150 Parish Drive
Wayne, NJ O7474 USA

and

William G. Swavely and C. Richard Unkle
IIT Research Institute
201 Mill Street
Rome, NY 13440 USA

ABSTRACT - The thrust of the project described is to integrate outputs of the System Testability Analysis Tool with a board level tester to achieve an optimal GO-NO GO test strategy and fault diagnostics to minimal component identification. The DETEX Systems, Inc.'s System Testability Analysis Tool (STAT) provides information on testability characteristics of a design. It can also be used to develop optimal fault isolation test strategies in the form of diagnostic flow tables. These tables provide an optimal test performance order to achieve fault-isolation based on topological design characteristics and one or more weighting criteria (e.g., reliability, test cost/time, component replacement cost/time). In an effort to take advantage of the test strategy outputs of testability analysis tools, GEC Marconi has enlisted the aid of IIT Research Institute (IITRI) to develop the means to control test execution order on a hoard-level Automatic Test System (ATS) based on the test order produced by DETEX's STAT.

This paper will discuss the objectives of this program, and present details on the results. Currently, a test program set (TPS) exists for an analog card that is being tested on a Sigma Series board-level test station. Software has been developed that will take a STAT diagnostic output report and develop *a* test program database. This database has been integrated with a TI'S wherein a test technician is automatically directed through probing sequences for fault-isolation. The database can also be used in an interactive maintenance aid format or as a test executive when probing is not required for fault isolation. Fault-isolation call-outs are also provided by the STAT, and this information is available during test. Details of the project progress, and any lessons learned will be provided.

I. INTRODUCTION/ BACKGROUND

Basing test program set (TPS) test order, both performance and diagnostics, on the outputs of model based analysis tools has been successfully demonstrated in the past [1], [2]. These tools have also been integrated with the tester such that the diagnostic database provided by the model and test selection algorithms dynamically choose the next best test based on available information from the test process. However, no such capability has been demonstrated for the System Testability Analysis Tool (STAT) produced by DETEX Systems, Inc. STAT does interface directly with test development productivity tools, such as TYX Corporation's PAWS, but the PAWS tools produce TPSs targeted primarily for ATLAS based test systems. An automated STAT link for other kinds of testers, such as the Sigma Series tester to be described in this paper do not currently exist.

GEC Marconi has continuously attempted to stay abreast of developments in testability that may benefit its production engineering group. In 1991, GEC Marconi began to evaluate a number of system level testability analysis tools, including DETEX's STAT, and the US Navy's Weapon System Testability Analyzer (WSTA). Based on these analyses, GEC Marconi is currently using STAT as its primary testability analysis tool. Other tools, such as Giordano Automation's Diagnostic Profiler^{TM} and Diagnostician^{TM} continue to be evaluated.

The tools being used by GEC Marconi provide a number of capabilities, but to date, they have been used primarily for testability analysis. In this capacity, the STAT provides the following information for a given set of tests and/or test points:

0-7603-2621-0'95 53.00 1995 IEEE

- Identification of any/all component ambiguity groups and the components within

- Percent fault-isolation to a specified ambiguity group size

- Relative fault-detection coverage percentages based on functional test

- Tests/test points NOT required to meet testability goals

- Identification of any/all feedback loops

In addition to the above testability analysis results, the STAT also provides optimal fault-isolation test strategies based on the topology of the design, component reliability, test cost/time, test point accessibility and other factors. It is this capability of STAT that GEC Marconi wanted to take advantage of that prompted the project to be described. Additional information on STAT, and other tools based on similar input can be found in reference [3].

II. PROJECT BACKGROUND AND OBJECTIVES

Currently test, and specifically diagnostics (i.e., faulty component identification) are cost drivers in the production operations environment. Another issue is the need for concurrent engineering methods to influence product designs to incorporate testability and Design for Testability (DFT) concepts to support lower cost test approaches. To address these issues, GEC Marconi, in conjunction with the Air Force Cost Reduction Initiative (CRI) Program, funded further enhancements beyond the baseline STAT tool set. The CRI program is designed to cost share with contractors to develop techniques/technologies that position both the military and contractors, such as GEC Marconi, to develop and deliver lower cost/more timely products.

The project selected addresses the use of a STAT circuit card model transitioned to an existing ATS (Sigma Series ATS) to validate the process of translating test flow strategy to production test equipment. Resulting from this endeavor should be the pathway to port testability analysis data to tester execution and specify rules to secure this pathway universally. The objectives of the project, therefore, was to demonstrate that the diagnostic outputs of STAT, in the form of a fault-isolation tree, could be used to "drive" the order in which tests are performed on an ATS, in an automated fashion, The eventual goal is to perform an analysis with STAT, select the test points required to meet testability goals, and then develop TPSs based on this analysis.

Previous experience with dependency model based tools has shown that test order can effect test efficiency, with the benefit of increasing TPS accuracy, decreasing TPS runtimes, and decreasing the overall cost of developing a TPS [2], [4]. The cost savings come primarily from the fact that test order, and fault-isolation call-out data are automatically provided by analysis tools, saving the test engineer time, while allowing concentration on test development.

There are currently a number of test automation tools, such as TYX Corporation's PAWS, and LCTI's LEXSYS that take the diagnostic outputs of STAT and/or WSTA, then use the test order produced in the development of both a Test Requirements Document (TRD) and TPS. Although these tools provide most of the integration between STAT and TPS development desired. these tools produce test code for primarily ATLAS-based testers. The system being used for this project, as well as others at GEC Marconi, are not being tested on ATLAS-based testers. Therefore, an alternative solution was desired.

Facilitation of the project was provided from following facts:

- An existing STAT model for the Analog Circuit Card Assembly (CCA) of the GEC Marconi Longbow System

- An existing performance TPS, without diagnostics for the Analog CCA

The existing STAT model was originally developed with assistance from IITRI, and therefore they were familiar with the Analog CCA design. The TPS was written independent of any analysis provided by the STAT. Further, the TPS is being hosted on a Sigma Series tester, The software for this tester is hosted on a PC, and runs under Microsoft Windows.

111. APPROACH/CURRENT RESULTS

Fig. 1 represents the approach that was taken to integrate the STAT diagnostic outputs with the TPS software for the Analog CCA. Although a model of the UUT already existed, the capability to convert a netlist of the UUT, captured in one of several formats such as EDIF, OrCAD[1], or Mentor Graphics, into a STAT model

[1] OrCAD is a registered trademark of OrCAD, Inc.

is now possible. From there, a complete STAT analysis can be generated, including the STAT diagnostic table, that contains an ordered list of tests, based on pass/fail .results, and a list of all resulting ambiguity groups that are isolated. An example of the STAT diagnostic table is provided in Fig. 2. Note in the table that when a fault is isolated, a Suspect Ambiguity Group (SAG) reference number is listed, along with the components that are contained within the group. The actual listing of the group, found after the diagnostic table in the STAT output, contains more detailed information. In addition to the isolations shown in the diagnostic table, it is possible to have STAT provide the suspect components that remain after each test step. In other words, if the first test shown in the table in Fig. 2 passes, this would eliminate some components from consideration. The components that remain as suspect are listed in the STAT output. Likewise, if test T19-2 (performed if T19-1 passes) passes, the components that are now suspect can also be found in the STAT output. Therefore, at any point in the diagnostic table one can determine which components remain as possible causes of system failure.

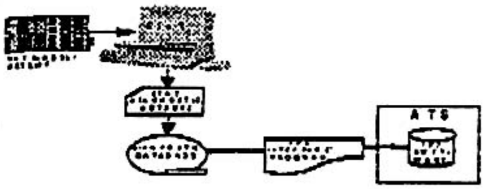


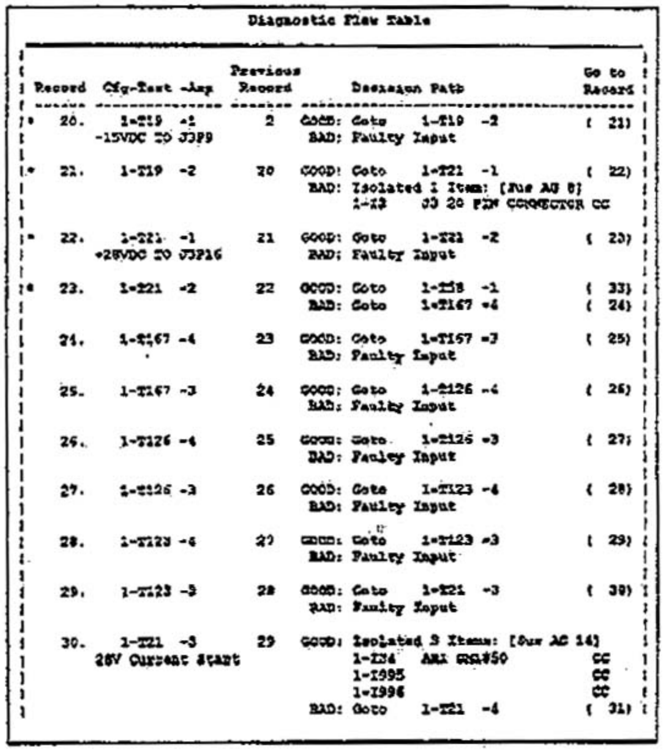
Fig. 1 Approach.

A. Integrating STAT with the TPS

The diagnostic database shown in Fig. 1 is created using a tool developed by IITRI that parses the information contained in a STAT diagnostic report into a database format. The tool was Written in Microsoft FoxPro for Windows[2], chosen for its speed, platform (Windows), and quick development time. The required input for the tool is a STAT testability report that must contain the following options:

1) Diagnostic Flow Table
2) Diagnostic Flow Comments
3) Suspect Ambiguity Group (SAG) Table
4) Diagnostic Flow Diagram ~~(optional)~~

Vito, "exclude SAG ..." option in reports must also be turned off
Rich



```
                         Diagnostic Flow Table
 ________________________________________________________________________
|                                                                        |
|                           Previous                               Go to |
| Record  Cfg-Test -Asp     Record      Decision Path              Record |
| ------  --------------   -------- ------------------------------------- |
|*  20.    1-T19  -1           2   GOOD: Goto     1-T19  -2        (  21) |
|        -15VDC TO J3P9              BAD: Faulty Input                    |
|                                                                        |
|*  21.    1-T19  -2          20   GOOD: Goto     1-T21  -1        (  22) |
|                                    BAD: Isolated 1 Item: [Sus AG 8]     |
|                                         1-X3     J3 20 PIN CONNECTOR CC |
|                                                                        |
|*  22.    1-T21  -1          21   GOOD: Goto     1-T21  -2        (  23) |
|        +28VDC TO J3P16             BAD: Faulty Input                    |
|                                                                        |
|*  23.    1-T21  -2          22   GOOD: Goto     1-T58  -1        (  33) |
|                                    BAD: Goto     1-T167 -4        (  24) |
|                                                                        |
|   24.    1-T167 -4          23   GOOD: Goto     1-T167 -3        (  25) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   25.    1-T167 -3          24   GOOD: Goto     1-T126 -4        (  26) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   26.    1-T126 -4          25   GOOD: Goto     1-T126 -3        (  27) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   27.    1-T126 -3          26   GOOD: Goto     1-T123 -4        (  28) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   28.    1-T123 -4          27   GOOD: Goto     1-T123 -3        (  29) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   29.    1-T123 -3          28   GOOD: Goto     1-T21  -3        (  30) |
|                                    BAD: Faulty Input                    |
|                                                                        |
|   30.    1-T21  -3          29   GOOD: Isolated 3 Items: [Sus AG 14]   |
|        28V Current Start                1-I34    AMI GRA#50        CC   |
|                                         1-I995                     CC   |
|                                         1-I996                     CC   |
|                                    BAD: Goto     1-T21  -4        (  31) |
|________________________________________________________________________|
```

Fig. 2 STAT Diagnostic Table.

This is the default start-up selection criteria. ~~The main screen contains option boxes to choose any of the above four choices. At this time, the tool requires that everything except the Diagnostic Flow Diagram be selected~~ before parsing. If isolation to a SAG at ANY TIME is required, option 4 above must also be selected. Otherwise, only final conclusions (i.e., fault isolation based on performing all tests in a specific diagnostic sequence), are provided.

The user extracts the required information from the STAT output by "parsing" the output file from within the IITRI tool. A "Parse STAT File" button allows the user to select the desired STAT output file, after which a descriptive name (20 characters) and a descriptive paragraph may he entered. The tool divides the STAT output into four relational databases which provide the GO/NO-GO flow (see Fig. 2).

Once a STAT has been parsed, the tool may be run in two modes, manual or automatic. In the manual mode, test steps, with optional test descriptions, are presented to the user. The user then performs the test and, based on test outcome, selects "Pass" or "Fail" to manually step through the diagnostic table until a SAG is isolated. Upon isolation, the Suspect Ambiguity items are listed to the screen. The user can also step backwards through

[2] Microsoft, FoxPro and Windows are registered trademarks of Microsoft Corporation

the tree to a prior test to, for example, change an answer from pass to fail or, if option 4 above was selected, the user can choose "Isolate Now" at any time to view the current items that are still suspect. This mode provides the capability to create a portable maintenance aid. Future enhancements are being considered that will integrate graphics and documentation that can be displayed using a hypertext based format. Test descriptions can also be entered if desired, either directly into the diagnostic database, or they are parsed and displayed in a description window if this information is entered into the STAT model. Such information would be contained in option 2 above, Diagnostic Flow Comments. Future use of this tool will make full use of this option in STAT.

The automated mode of using the diagnostic table information is explained in the following subsection.

B. Analog Card Testing

The ATS being used to test the Analog Card consists of six power supplies (4 x 30V @ 10A, 2 x 60V @ 5A) and a two frame Sigma Series Tester. The Sigma Series Tester is configured with 335 high speed digital pins (I60 Inputs, 160 Outputs, 15 Handshaking), 32 open collector digital power pins, a function generator, a two channel counter/timer, and a 6 1/2 digit DMM. Switching is handled through 40 channels of low loss analog switches, 30 channels differential scanner, and a seven channel high frequency matrix. The interface adapter contains circuitry to generate multiple clock signals, the necessary loading to simulate motors, and a filter circuit to enable measurement of the motor drive outputs generated. by the Analog Card.

As previously mentioned, a TPS used to verify card functionality existed prior to project initiation. This TPS did not, however, contain any diagnostic routines. As part of the project then, a diagnostics subroutine was added. to the existing TPS that would interface with the diagnostic database described above using files for handshaking/communication. Three files are currently used. The first file, named RESULTS.LOG, is written by the ATS and initially contains the results of the performance test. Upon detection of the first failure, the RESULTS.LOG file is passed to the database tool, where the first diagnostic test for the failed performance test is selected. The other two files, called NEXTTEST.OUT and SAG.OUT are written by the diagnostic database tool. The NEXTTEST.OUT file will contain the name of the diagnostic test to be performed, using the STAT generated labels as shown in Fig. 2 (E.G., T279-1). The SAG OUT file will contain a listing of the suspect component(s) that may have caused the failure being diagnosed. Additional details of each file are provided below.

Upon failure of a performance test, the diagnostic subroutine sets a flag identifying that diagnostics are now being performed. The failed test name is written to the RESULTS.LOG file that is subsequently read by the diagnostic database tool. Based on which test failed, the database tool either writes a result to the SAG.OUT file, or the first diagnostic test to the NEXTTEST.OUT file. The diagnostic subroutine then reads a test name from the la EXTTEST.OUT file as described above, and calls the appropriate subroutine for that test. The existing TPS subroutines were modified to use the diagnostic flag to determine whether to call the test executive (to compare and output data) or a probe routine (with information on where to probe and what to look for). Presently, probing is done with an oscilloscope and the aid of a pop-up window containing a description of the test to be performed (e.g., "Is there a 28 volt square wave on U9 pin 3 ?" Yes/No). Probe results are written to the RESULTS.LOG file, which is then read by the diagnostic database as before. The database tool then selects another test, or isolates to an SAG. The SAG information is then written to the file SAG.OUT which contains a listing of the components contained in the suspect ambiguity group. This information will then be used for generating the rework procedure.

IV. FUTURE CONSIDERATIONS

At the point in time when this paper was being written, a complete demonstration of the automated link between the TPS and STAT diagnostic database was not performed. However, no problems are anticipated based on simulations that proved out the capability of having the STAT diagnostic database tool communicate with the Sigma Series Run Time Software. Once the integration is completely demonstrated, future plans are to develop a probe routine that uses the system resource and a foot switch (i.e. "Place probe on U9 pin 3 and press foot switch"). This will eliminate the need for the test engineer to determine the goodness or badness of a test. Under this scenario, the software will automatically make the determination, and write the result to the RESULTS.LOG file. Other considerations for future work include developing enhancements to the STAT parsing tool developed by IITRI as described, and, if necessary, adding any enhancement required to optimize the speed of test execution.

Given that this demonstration is successful, GEC Marconi has plans to use these tools to aid in the testing of additional Analog Cards, and other systems developed by GEC Marconi. The expected benefits are decreased test development time and increased accuracy in fault diagnostics. Finally, plans are being made to automatically capture the SAG information into a logistics database that can be used to track overall reliability performance of the items being tested. This information can then be fed back to the STAT tool and, if desired, a new test strategy can be developed that takes into consideration actual performance data.

REFERENCES

[1] Li Pi Su, G. De Marc and M. Nolan, "DARTS: An Enabling Technology fox Concurrent Engineering;" Proceedings, AUTOTESTCON '93, San Antonio, Texas, September 1993.

[2] Harr H. Dill, “Test Program Sets - A New Approach,” Proceedings, AUTOTESTCON '90, San Antonio, Texas, September: 1990.

[3] “Testability Design and Assessment Tools," Reliability Analysis Center Report CRTA-TEST (by G.R. Unkle„ prepared under contract to Rome Laboratory, Griffiss APB, NY

[4] F. Johnson and R. Unkle, "The System Testability And Maintenance Program (STAMP"); A Testability Assessment Tool For Aerospace Systems," Proceedings, AIAA/NASA Symposium on the Maintainability of Aerospace Systems, Anaheim, CA, July 1989.