INTEGRATING DESIGN FOR TESTABILITY AND AUTOMATIC TESTING APPROACHES

Eugene A. Esker,[1] Jean-Paul Martin,[2] William R. Simpson,[1] and John W. Sheppard[1]

[1]ARINC Research Corporation
2551 Riva Road
Annapolis, Maryland 21401

[2]Avionics and Systems Division
Aerospatiale
Toulouse, France

ABSTRACT

Current approaches to developing automatic test equipment (ATE) for avionics systems are resulting in many different ATE architectures and configurations. Because of this variety, maintenance shops typically must own a test station for each system to be tested; therefore, these test stations are not fully used. The Standard Modular Avionics Repair and Test (SMART™) concept was developed to provide an approach to standardizing ATE architectures and modularizing the ATE and test program set (TPS) development process. The System Testability and Maintenance Program (STAMP®) was developed to meet the need for designing testable systems and verifying the testability of systems. The Portable Interactive Troubleshooter (POINTER™) was developed as an intelligent test executive to dynamically manipulate testability models. Using the SMART approach to automatic testing and the STAMP approach to design for testability, an approach for integrating the two with a dynamic tool—POINTER—into a complete architecture for ATE and TPS development is presented.

INTRODUCTION

With the increased emphasis on modular architectures and portable software, current approaches to developing automatic test equipment (ATE) are not meeting the needs of the users for a common test set that maintains their collection of avionic units. Test stations often are developed by line replaceable unit (LRU) vendors and are structured around the specific LRU being tested. Consequently, maintenance shops frequently must have individual test stations for each system, resulting in underutilization of these assets.

Aeronautical Radio, Inc. (ARINC), is addressing the need for a standard approach to automatic testing by designing a modular architecture for automatic test equipment. ARINC has developed the Standard Modular Avionics Repair and Test (SMART™), which includes:

- A suite of instruments, a standard switching device between the instruments, and an LRU-specific test unit adapter (TUA)
- Portable ATE system software running on the test control computer
- A transportable test program for the unit under test (UUT)

Further, the test program is developed by using several standard libraries that describe the available resources (necessary for resource allocation), the specific devices, and characteristics of the TUA.

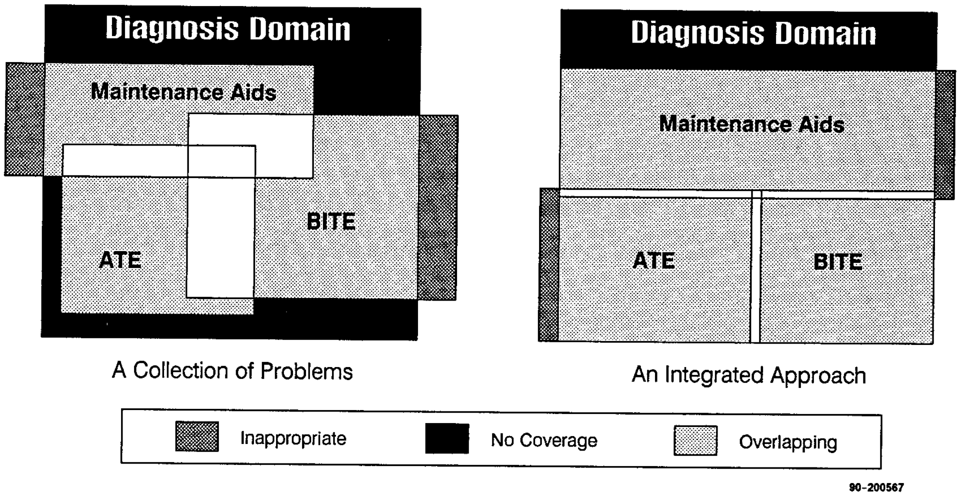
Independent of test station design and important to the task of developing an effective and efficient automatic testing capability is the need to design for testability. Identifying the inherent testability of a system design is essential to determining limitations and pitfalls that would otherwise arise in maintaining the fielded system. Once limitations are identified, the design can be modified (if necessary) to correct the problems, and strategies can be developed to effectively use the inherent testability to its fullest. Figure 1 depicts the current approach and an integrated approach to diagnosis, showing the improvement of treating the ATE as part of an integrated design (i.e., reduced overlap and inappropriate work).

In this paper, we describe the SMART architecture and an approach to incorporating the concept of design for testability and maintenance aid tools to develop standard, flexible, and effective ATE systems. We also discuss how this integrated approach can provide a complete, integrated maintenance architecture and how to extend the approach to provide an intelligent ATE system by incorporating some of the analysis techniques in the ATE architecture.

INTEGRATED MAINTENANCE

Integrated maintenance takes into account the testability of a system, the approach to diagnosing a malfunction, accurately reporting the failure across hierarchical bounds (i.e., between levels of maintenance), and providing access to the repair procedures. System designers must practice a disciplined approach to integrated maintnenace to apply it successfully. Because today's functional systems are more complex than those of a generation ago, the need for this integrated approach becomes driven by the increased complexity of technology.

CH 2793-8/90/0000-0509 $1.00 © 1990 IEEE



A Collection of Problems

An Integrated Approach

FIGURE 1. EFFECTIVE TESTABIITY IMPROVES SYSTEM PERFORMANCE

Integrated maintenance comprises three elements: built-in test equipment (BITE), ATE and ATE test program sets (TPSs), and interactive maintenance aids. The "Collection of Problems" portion of Figure 1 depicts how well BITE, ATE, and maintenance aids currently cover the diagnostic domain. BITE is usually provided by the airborne electronics designer. Generally, ATE (and its TPS) and portable maintenance aids (if necessary) are produced by the LRU or system vendor. The inefficiencies evident in Figure 1 include areas that lack coverage, areas of overlapping coverage, and in some cases, maintenance actions residing outside of the diagnostic domain altogether. The "Integrated Approach" of Figure 1 depicts a closer alignment of the three maintenance elements.

The need for an integrated approach to maintenance is now well accepted. The Joint Integrated Avionics Working Group (JIAWG) has produced standards for advanced avionics architectures incorporating integrated maintenance concepts.[1] The Future Concepts for Maintenance (FCM) Task Group of the Avionics Maintenance Conference (AMC) is specifying architectures for on-board maintenance and automatic test. [2,3] The Department of Defense has even developed a standard for testability assessment of electronic systems.[4]

Of primary concern is the automatic testing portion of the integrated maintenance concept. Reference 5 describes a fully integrated approach to system maintenance. The first point of concentration is the problem of standardizing the development of ATE and TPS, one necessary step in making the integrated maintenance concept a reality.

STANDARD MODULAR AVIONICS REPAIR AND TEST—SMART

The use of ATE, an important element of integrated maintenance, increases productivity, but many avionics systems are being maintained by dedicated test stations, resulting in high initial costs. To control costs, the airlines, through ARINC, have developed SMART, a modular architecture including a set of standards for a generic test system, which allows a freedom of choice in selecting test instruments, TUAs, and test control computers (TCCs). The SMART integrated software can be hosted on different development systems. The TPS can be targeted to a full range of test control computers from high-performance personal computers (e.g., 80386 microprocessors) to mainframes. The adaptable standard ATE is designed to drive and monitor most avionic units regardless of the manufacturer.

The key features of SMART are depicted in the software flow diagram of Figure 2. Some major attributes of SMART are the following:

- *Transportability of the TPS:* Users can transport complete TPSs (compiled from ATLAS) from one SMART system to another, thereby reducing or eliminating multiple TPS development for common avionics in different aircraft.
- *Open System Architecture:* SMART can be sized to support any ATE constructed in accordance with ARINC Specification 608 from small dedicated systems to large, general-purpose systems.[3]
- *Freedom of Choice:* Users can select or implement hardware and operating systems of choice. Special interface to nonstandard items can be designed and implemented.
- *Modularity:* Users can tailor test systems to contain only those elements necessary for each job and retain the flexibility to reconfigure the test station as the job changes.
- *Software Portability:* SMART moves easily from one TCC to another because of its modularity and the predefined hardware and software interfaces. The software is written in the American National Standards Institute (ANSI) standard C language.

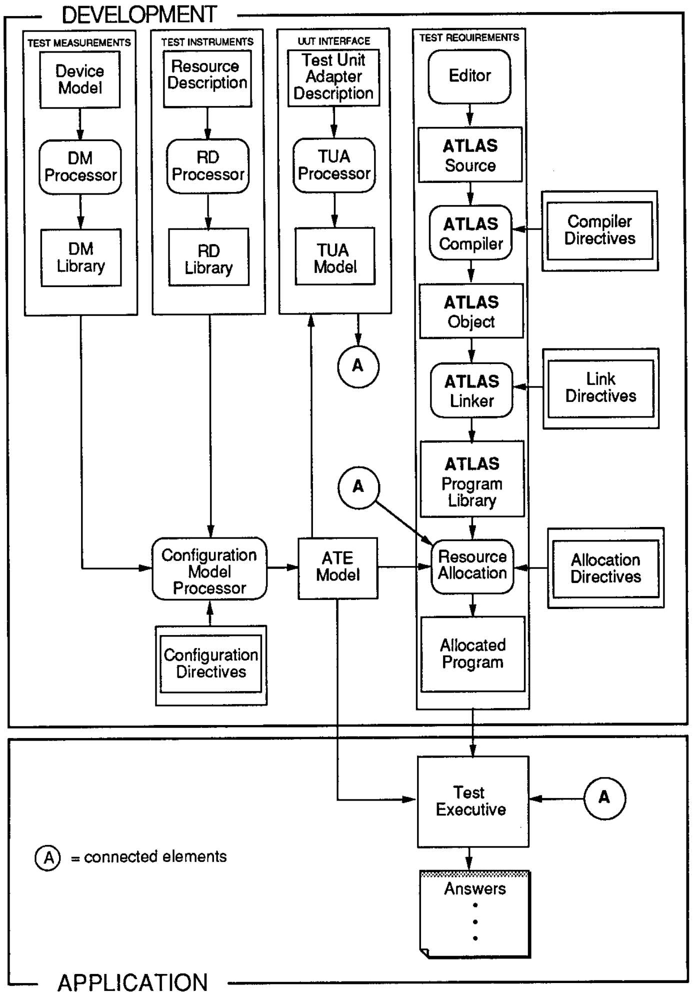


FIGURE 2. STANDARD MODULAR AVIONICS REPAIR AND TEST—SMART—SOFTWARE FLOW

TESTABILITY VALIDATION

Good testability is a key element in accomplishing integrated maintenance. Equipment has good testability when existing faults can be confidently and efficiently identified. MIL-STD-2165 defines testability as "a design characteristic which allows the status (operable, inoperable, or degraded) to be determined and the isolation of faults within the terms to be performed in a timely and efficient manner."[4] The literature identifies two types of testability when referring to system-level testability—inherent testability and achieved testability:

- ***Inherent testability*** is concerned with the way the system is designed and the ability to observe system behavior using a variety of stimuli. It is defined by the location, accessibility, and sophistication of tests and test points applicable to the system.

- ***Achieved testability*** is concerned with the way maintenance of a system is implemented. It is defined by the results of the maintenance process (e.g., false alarms, ambiguities, incorrect isolations, no faults found).

The testability analysis process should begin during the design phase. This permits maximizing a system's inherent testability. Beyond the design phase (and without the ability of accomplishing design changes), a testability analysis can only identify how to make use of the inherent testability more intelligently (i.e., achieved testability).

A testability analysis should identify:

- *Ambiguity Groups:* Failures not uniquely isolatable in the current configurations.
- *Information Feedbacks:* A group of test elements whose outcome or output is part of an information circularity, and thus a test depends upon all elements within the group, if it depends on any. Any test or failure element in the information circularity is part of the feedback.
- *Test Disposition:* This has two basic subgroups:
  - Tests not needed. This information can be used to reduce maintenance complexity and TPS test times.
  - Additional tests needed. This information is essential in improving testability.
- *Undetectable Component Failures:* Failures that are not observed by any of the available tests.
- *False Failures:* A combination of multiple failures that provide the same symptoms as does an unrelated single failure.
- *Hidden Failures:* The case where the root cause is masked by the observed symptoms.
- *Tolerance to False Alarms:* A flag to insert downstream checks to identify false alarms.
- *Operational Isolation:* The percentage of time expected to be needed to isolate to *n* or fewer replaceable units. This information is critical in verifying specification compliance and logistics planning.

Upon completion of the analysis, diagnosis is required to complete the testability validation. Diagnosis consists of three basic actions: 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 BIT or its associated equipment—BITE—and may actually be the design criterion set upon BIT and BITE.

Localization refers to the ability to say that a fault has been restricted to some subset of possible causes. Faults are localized using a combination of tests or a diagnostic strategy. All BIT localizes to at least one of all possible faults. If the localization is sufficient, in most cases, to undertake repair, we often refer to that BIT as "smart BIT. " BIT, however, is not the only diagnostic technique that localizes faults. Often ATE and manual isolation techniques employ diagnostic strategies that localize faults to a degree sufficient to undertake repairs.

Isolation means that through some test, combination of tests, or diagnostic strategy a specific fault is identified. It is often assumed to mean, however, that localization has been achieved to a degree consistent with a single repair unit.

An effective diagnostic strategy must provide a procedure that brings the achieved testability up to the level of the inherent testability. The point to be made here is that with the complexity of modern equipment and without the use of a good testability analysis tool, implementation of the fault-isolation process often falls short of inherent testability. Inherent testability, as a design characteristic, is the upper limit of the effectiveness of the diagnostic strategy.

STAMP AND SMART —INTEGRATED TESTABILITY USING CURRENT TOOLS

Use of a testability tool to evaluate system testability hierarchically in combination with SMART architecture can achieve integrated maintenance. Many tools exist but few are hierarchical,[6] and fewer still can handle a system of any complexity. One tool that meets these criteria and that has been tested in more than 50 systems is the ARINC System Testability and Maintenance Program (STAMP®).[7] This combination results in powerful ATE systems with efficient and effective TPSs for fault isolation.

STAMP is a computer-aided testability analysis and fault-isolation development tool. STAMP measures testability and synthesizes fault-isolation strategies on the basis of an information flow model of the system (or unit) under analysis. The techniques have been described in detail in references 8 through 10.

STAMP analysis has been applied to more than 50 systems,[7] 1 of which consists of more than 120 LRUs. Each application either has resulted in improvements in field performance or has been used in early design phases to define the testability characteristics. It is clear from these results that designing adequate testability into equipment and designing and executing TPSs to make use of the testability is an effective and economical approach to integrated maintenance.

A primary STAMP output is its fault-isolation strategy (fault tree) optimized by information value of the tests as well as several other criteria (e.g., test times or failure rates). The point of interface between STAMP and SMART is through the fault tree that forms the basis of TPS development. The fault trees define the sequence of testing the TPS will follow to fault-isolate the UUT. STAMP fault trees have proved to be efficient and effective for fault isolation.

ACHIEVING AN INTELLIGENT ATE

The development of testability models of UUTs can result in more intelligent ATEs through the dynamic manipulation of these models. By including an inferencing capability, we can optimize the testing strategy using information learned through status indicators or prior tests. The Portable Interactive Troubleshooter (POINTER™) software, developed by ARINC Research Corporation, dynamically manipulates testability models.

POINTER is a portable, intelligent, troubleshooting system that provides maintenance technicians with on-site capability to quickly diagnose equipment failures. POINTER may be expanded to include on-line documentation, training, logistics analysis, and learning functions. POINTER may also be used as an embedded system for smart BIT, configured off-line as an intelligent ATE controller, or applied to other configurations. POINTER requires data in the form of component and test dependencies as generated by STAMP. The POINTER system is described in greater detail in reference 11.

The elements necessary for an intelligent ATE are identified and can be integrated over the next several years. Figure 3 shows the architecture by which an interface between SMART, STAMP, and an intelligent test executive such as POINTER can be developed. Figure 3 should be compared with Figure 2, which is the current existing SMART architecture. Note that the TPS development phase has been replaced by a system modeling phase and a test requirements library development phase. The system model provides the knowledge base for a POINTER inferencing capability. The test model library differs from a TPS in the following ways:

- Each test must stand alone, including a definition of machine set up for the test (modular tests).
- No sequencing of tests is provided or implied.
- The collection of tests is executable in any sequence.

By incorporating an intelligent test executive, current information, including failure history, on-board BIT or on-board maintenance system (OMS) data (that placed the UUT on the test equipment), and operator input, forms a context about which test choices are made and executed. The intelligent test executive computes the test sequence each time a UUT is tested to achieve optimal efficiency. Although there is an overall similarity in Figures 2 and 3, small architectural changes can accomplish a large potential gain achievable through a modular approach to an intelligent ATE.

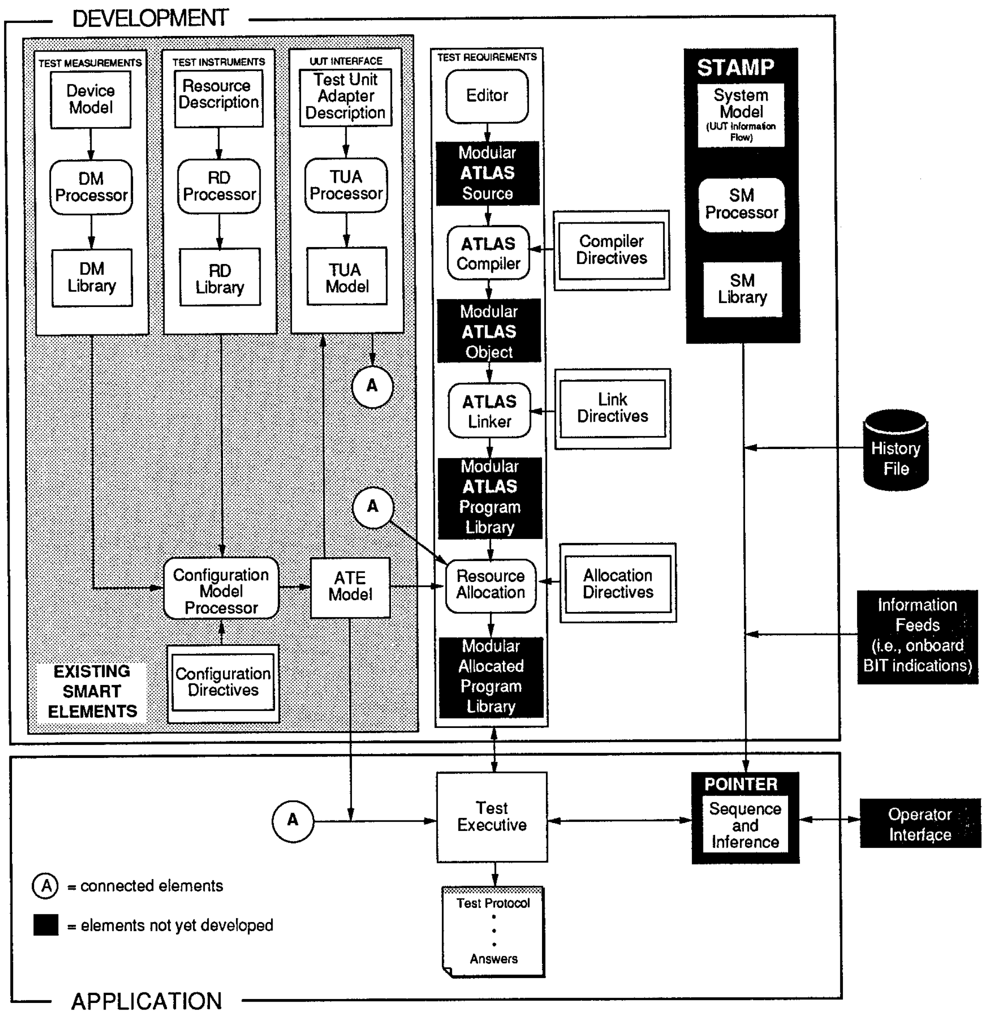


FIGURE 3. MODULAR APPROACH TO INTELLIGENT ATE

SUMMARY

A standard, modular approach to ATE systems and TPS development that uses SMART architecture is currently being implemented for the airline industry. A major consideration is to design testability into a system and take advantage of the resulting inherent testability to define effective troubleshooting procedures. This analysis is accomplished by using a tool such as STAMP. It is possible to integrate the STAMP process with the SMART architecture.

SMART can also become an intelligently controlled ATE and still retain its essential features of hardware independence and modular repair and test. This can be accomplished by integrating SMART with STAMP and by using POINTER as an intelligent test executive to dynamically manipulate the testability model. Although the tools currently exist, software architecture needs to be developed to achieve this union.

REFERENCES

1. Joint Integrated Avionics Working Group, *JIAWG Advanced Avionics Architecture Standard J87-01 (CAB II)*, Dayton, Ohio, January 1989.

2. Aeronautical Radio, Inc., *Onboard Maintenance System,* Draft 3 of Project Paper 624, working paper of the AEEC, Annapolis, Maryland, April 1989.

3. Aeronautical Radio, Inc., *Standard Modular Avionics Repair and Test Systems (SMART™),* ARINC Project Paper 608, Annapolis, Maryland, September 1989.

4. 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.

5. J. W. Sheppard and W. R. Simpson,, "Integrated Maintenance—A Hierarchical Approach," *AUTOTESTCON '90 Conference Record,* San Antonio, Texas, September 1990.

6. D. Buczek, "Some Testability Models with Brief Data," STAMP Technical Note 650, ARINC Research Corporation, Annapolis, Maryland, March 1987.

7. R. Unkle and F. Johnson, "The System Testability and Maintenance Program (STAMP): A Testability Tool for Aerospace Systems," *Proceedings of the 1989 AIAA/NASA Symposium on the Maintainability of Aerospace Systems,* Anaheim, California, July 1989.

8. W. R. Simpson, "Testability and Fault Diagnosis of Airline Avionics," Special Edition of *PLANE TALK,* AMC Open Forum, Houston, Texas, March 1983.

9. W. R. Simpson, "The Application of the Testability Discipline to Full Systems Analysis," 1983 IEEE Automatic Test Program Generation Workshop, San Francisco, California, March 1985.

10. W. R. Simpson, "System-Level Diagnostics: Simplifying Complexity," 1989 IEEE Workstation Symposium, Johns Hopkins University, Laurel, Maryland, October 1989.

11. W. R. Simpson, J. W. Sheppard, and C. R. Unkle, "POINTER—An Intelligent Portable Maintenance Aid," *AUTOTESTCON '89 Conference Record,* Philadelphia, Pennsylvania, September 1989.