TM-3143-1717
30 SEPTEMBER 1987

TESTABILITY ANALYSIS TOOLS ON A MILITARY SYSTEM

TECHNICAL REPORT

RESEARCH AND DEVELOPMENT

**TEST TECHNOLOGY
INFORMATION CENTER
AND EVALUATION**

prepared for
NAVY TEST TECHNOLOGY CENTER

prepared by
NAVAL OCEAN SYSTEMS CENTER

NAVAL WEAPONS STATION, SEAL BEACH, CORONA ANNEX
FLEET ANALYSIS CENTER
CORONA, CALIFORNIA

FOREWORD

1. The Test Technology Information Center (TTIC) is chartered by SPAWAR to acquire, organize, store, and disseminate Research, Development, Test, and Evaluation (RDT&E) information on a Department of Defense-wide basis.

2. This document reports the results of the author actually modeling and running several testability figure of merit (TFOM) tools on a benchmark mixed technology complex system. This work and report were performed under TTIC Work Request N6070187 WR00086AA in support of the SPAWARS Test Technology Office.

3. Additional copies of this report or new reports on other areas of test technology may be obtained by telephoning (AV)933-5451 or commercially (714)736-5451. Written inquiries should be forwarded to:

Navy Metrology Engineering Center
Navy Weapons Station, Seal Beach, Corona Annex
Attn: Test Technology Information Center (Code 3143)
Corona, CA 91720-5000

TESTABILITY ANALYSIS TOOLS
ON A MILITARY SYSTEM

30 SEPTEMBER 1987

TABLE OF CONTENTS PAGE

LIST OF ILLUSTRATIONS PAGE

INTRODUCTION

The author has published two earlier government reports on Testability Figure of Merit (TFOM) tools used on Small Scale Integration (SSI) and Large Scale Integration (LSI) Printed Circuit Board (PCB) level models. It is obvious that most TFOM tools have been intended and utilized at the IC chip or PCB level, but little information can be found about TFOM applications at the higher system level. Since government contractors must comply with the testability Military Standard (MILSTD) 2165 at the system level as well as the module level, industry and government need the system test hierarchical level to be addressed.

The job can be broken into several distinct but interrelated tasks. First, all of the potential TFOM tools that could be used had to be identified. Next a determination as to system level (preferably to cover more than just digital logic) applicability had to be made. A list of TFOM tools used, with explanation of their acronyms is provided in Table 1. Finally a system of mixed digital, mechanical, and analog technologies had to be located and modeled. After those three tasks, it was a matter of redoing the system model for each unique TFOM entry and getting desired output reports. The final task is recording what happened on each of the eight TFOMs in this report.

Four of the TFOM tools used in this report have been used and reported on by the author previously at the PCB level. These are MILSTD-2165, LOGMOD, STAMP, and Daisy Testability Analyzer (DTA). The other four TFOMs are just available in 1987, and some were still in beta testing and developing user manuals in mid-1987. The Architecture Design and Assessment System (ADAS) software by Research Triangle Institute (RTI) was initially to be included in this report but was eliminated when it became apparent that it had no testability outputs. Frankly, one TFOM tool included in this report was really only useable at the digital chip PCB level, with negligible system capability. That was the DTA, but it was included as a trial to see if it is feasible for users that have invested in costly CAE workstations and digital chip libraries, to use these assets in system TFOM analysis. Many of the vendors said this was the first system level application of their various TFOM tools.

1.1 APPROACH

The choice of a proper benchmark system was crucial and the M-1 tank Fire Control System (FCS) met all of the parameters desired for a good system exercise of the TFOM tools. It included modules or boxes with digital, analog and mechanical technologies. Many of the TFOM software tools and graphics were obviously designed for the digital device and PCB level analysis, but innovation and work-arounds made them function at a higher level than intended. This FCS system can be modeled at a simple box flow in PCs that are graphic and memory limited to one screen, or at a detailed multi-screen CAE level as well.

TOOL	ACRONYM MEANING
ASTEP	ADVANCED SYSTEM TESTABILITY EVALUATION PROGRAM
CAFIT	COMPUTER AIDED FAULT ISOLATION & TESTABILITY
DTA	DAISY TESTABILITY ANALYZER
IN-ATE	INTELLIGENT AUTOMATIC TEST EVALUATION
LOGMOD	LOGIC MODEL
SCOAP	SANDIA CONTROL-OBSERVABILITY ANALYSIS PROGRAM
STAMP	SYSTEM TESTABILITY ANALYSIS MAINTENANCE PROGRAM
ACE	APT COMPUTATIONAL ENVIRONMENT

Table 1 TFOM tool acronyms for algorithms used in system level analysis.

COMPONENT	MTBF PREDICTION	FIELD DATA
LOS EU	15,818	5,183
GPS PANEL	102,470	154,196
TNB	4,659	4,196
GTD EU	5,308	5,458
GUNNER HANDLE	118,050	3,009
COMMANDER HANDLE	314,564	13,706
AZIMUTH SERVO	40,634	6,047
ELEVATION SERVO	16,250	6,778
GUN TRUNNION RESOLVER	1,776,199	26,817
CMDR CONTROL PANEL	93,345	38,549
LOADERS PANEL	19,552	616,784
COMPUTER ECU	150,000	5,363
COMPUTER CONTROL PANEL	11,995	22,844

Table 2. M-1 tank FCS Mean Mile Between Failure (MMBF) predictions and field Mean Miles Between Replacement (MMBR) field data history. The latter was used for R&M inputs in this study.

Interconnect variations include single lines between boxes, all the way up to every signal wire between boxes being represented. The interconnecting cables were modeled as replaceable units with failure data on some models also.

A prime driver toward use of the M-1 tank FCS as the system benchmark, was the availability of a detailed and accurate logic model for this system at nearby General Dynamics Electronics (GDE) in San Diego. Not only did GDE's Lloyd Mills assist in scoping and recoding the system for this limited application, but GDE also provided access to their Daisy CAE workstations for nearly two weeks. Naturally, all TFOM vendors kindly allowed free access to their facilities and software, and provided assistance in the modeling efforts and assessment of the output results. Figure 1 shows the high system level block that overviews the system. Variations of this will be noted as entry models throughout the report sections. The tank turret FCS was split out from the total turret system and reduced down to 17 boxes and nine cables.

Figure 2 shows actual hardware units represented by some of the block diagram boxes. Abbreviations of tank FCS units are Computer Electronic Unit (CEU), Gun Primary Sight (GPS), Line of Sight Electronic Unit (LOS EU), Gun/Turret Drive (GTD), Gun Trunnion Resolver (GTR), Commander Control Panel (CCP), Commander Control Handle (CCH), and Gunner Control Handle (GCH). The Terminal Network Box (TNB) had five internal Circuit Breakers (CB) that were modeled as Lowest Replaceable Units (LRU), even though inside of a system block function. Other modeling tips are contained in Section 1.3.

Increasingly, more of the tools include reliability data input values, that usually help drive the algorithm to weight the less reliable areas for test. Some tools have Fault Isolation (FI) or portable maintenance aid outputs. This is beyond the scope of the up front Design For Testability (DFT) intent of this study, but such capabilities will be noted. It is true that Automatic Test Program Generators (ATPG) grade a design for test coverage and testability analysis, but test programs are normally not written until the design is firm and in production. ATPGs are written at the module level for in-circuit or edge pin functional Automatic Test Equipment (ATE), which is not at the system level in any case. This system level analysis would assume that DFT analysis is done separately for each module internally.

1.2 METHODOLOGY

The use of these TFOMs on a fielded system is a little artificial, since most tools are intended to advise on specific areas to modify the design early in the design process. The design was entered with no DFT changes and a TFOM was derived for the system, however on a couple of tools, the urge to play "what if" games could not be resisted. When additional Test Points (TP) were added, the resultant change in the test report outputs will be noted. Comparison of recommended design changes cannot be directly correlated due to differences in modeling and differing node or signal designations for various TFOM models.

One benefit of modeling a fielded system, not available to a new preproduction design, is the use of real failure data instead of the misleading Military Hand Book (MIL-HDBK) 217 predictions. As is shown in Table 2, there is a considerable variation between MIL-HDBK predictions and the real world field failure data for reliability. This inaccurate data not only degrades the TFOM and FI outputs, but disrupts the sparing and support policies for systems. The actual Army MTBF field data on the LRUs is used in this report. Due to the higher numbers having more weight on the TFOM algorithms, the MTBF numbers were recalculated to scale so that high failures had large values, which is the opposite of traditional reliability procedure. Table 3 shows the author calculated reliability weightings, scaled from 1 to 200 ratio to agree with the field failure feedback. Normally high failure devices have low number values.

When the TFOM graphics and memory could handle it, the cables between the boxes were modeled as failure items. The author's field repair experience showed cables and connectors tend to fail more often than the boxes or assemblies that they link. One week was estimated as the model entry and report generation time for all TFOMs for planning purposes. Some took only three days, and another required nearly two weeks. Certainly the travel and hotel costs created an artificial time factor probably not applicable to TFOM users working in their own facilities with intimate knowledge of their system design. The Section 2 detailed description will have each system TFOM modeling tool covered in a format structure of Background, Input, Output, and Comments. Since the system modeling requires a whole new outlook, methodology and discipline, a summary of lessons learned and system modeling approaches is included to help other future users.

1.3 SYSTEM MODELING APPROACHES/VARIATIONS

Table 4 shows three levels of detail of model entry for the eight TFOM tools, and modeling variations will be covered in the detailed section for each TFOM. Two TFOMs, STAMP and CAFIT, were modeled at two different levels which show the gains of more detailed data entry.

* Do not model below the Lowest Replaceable Unit (LRU) level, which is normally the PCB or plug-in module level.
* An exception could be if some PCB or unit components are normally replaced in the field, such as the five CBs in the TNB for this system. Some PC-based tools may have hierarchical limitations to embedded LRU modules.
* When several parallel wires or conductors have identical data or signals, these could be combined in one line, possibly with a "(5)" for clarity.
* Normally connectors, pins, or cables are not modeled as items, but if some connectors may open, or cables crimp, these should then be modeled on a case basis.
* In logic models, when an item has several outputs, it will appear several times in the report. When ready to fault analyze ambiguity groups, common "items" must be combined or collapsed into the "one" hardware item.

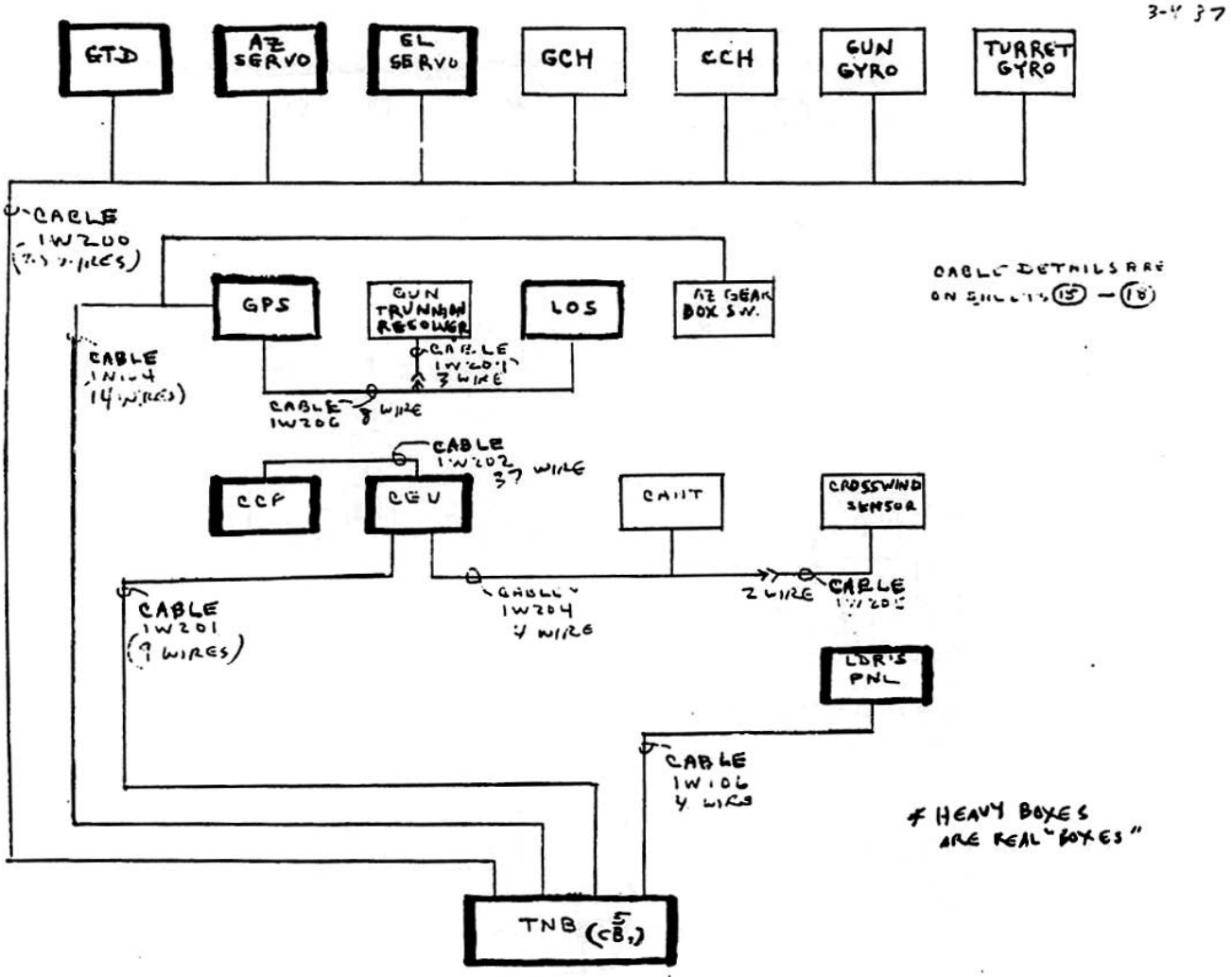

Figure 1. Basic block diagram of M-1 tank Fire Control System components used as the basic benchmark model for this report.

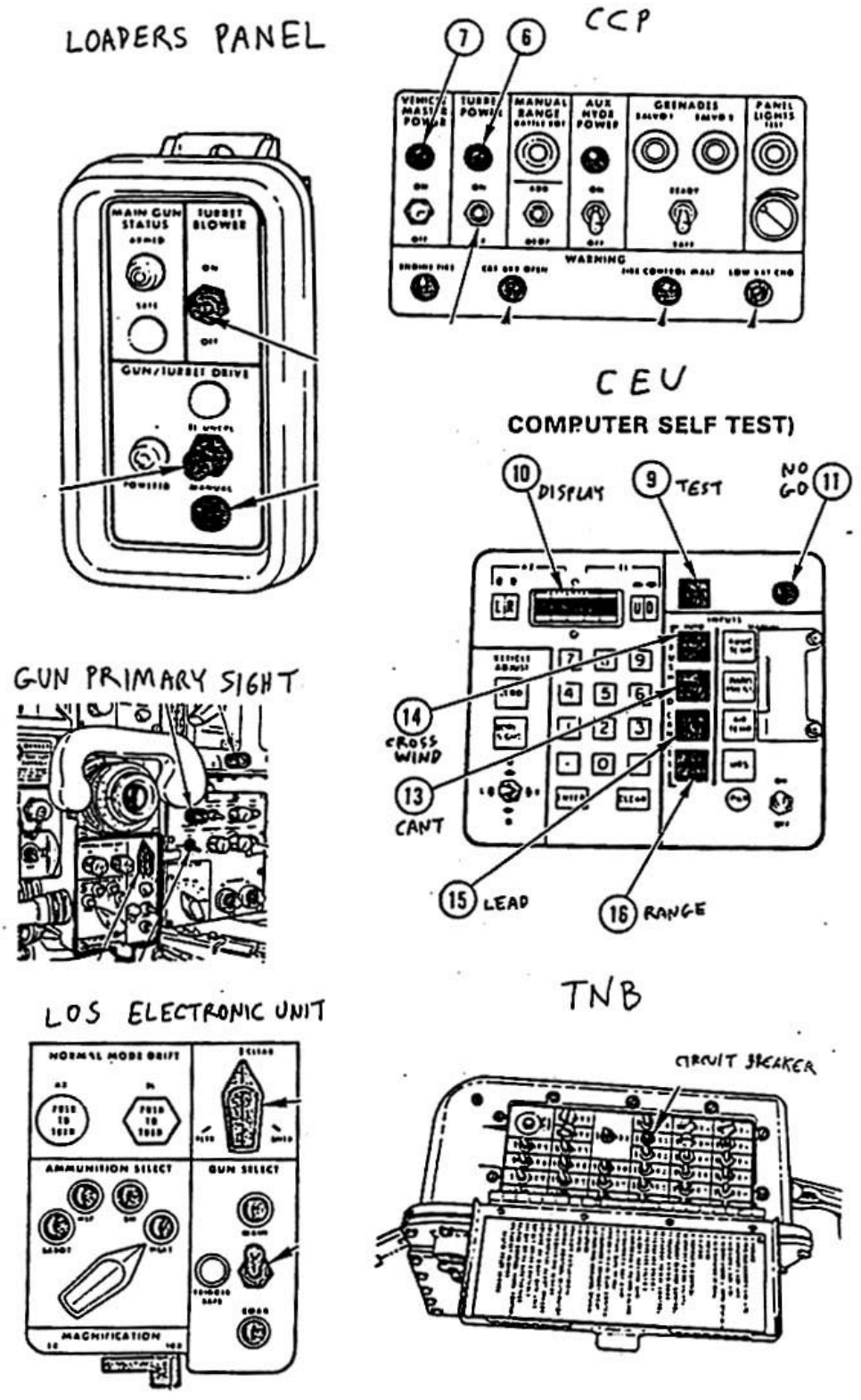

Figure 2. Appearance of several actual FC boxes from the M-1 tank, used in this system analysis.

COMPONENT	FIELD MMBR	FAIL WEIGHT	TEST COST	REPLACE COST
GUNNER HANDLE	3009	20,000	10	40
TNB	4196	15,400	40	120
LOS	5183	11,846	20	30
CEU	5363	11,492	2	30
GTD	5458	11,282	20	30
AZ SERVO	6047	10,266	8	40
EL SERVO	6778	9.059	8	40
CMDR HANDLE	13,706	4,496	10	40
COMP CONT PNL	22,844	2,702	15	30
GTR	26,817	2,298	20	30
GPS	154,196	400	15	20
LOADER PNL	616,784	100	10	20
TNB CKT BRKRS	2,000*	25,000	1	1
CABLES	5,500*	3,000	25	20
AZ GEAR SWITCH	6,500*	8,000	50	30
GYROS	6,000*	10,000	10	30

Table 3. Reliability data weighted for model entry. Estimated test costs and replacement costs for entry to TFOM tools sensitive to such information. Note * means estimated values.

TFOM TOOL	EVERY BOX IN/OUT	SOME BOX IN/OUT	MINIMAL BOX I/O	PC KEYBOARD	PC SCHEM	CAE SCHEM
LOGMOD	X			X		
STAMP	X		X	X	X	
A-STEP		X		X		
2165	-	-	-	-		
IN-ATE			X		X	
DTA	X					X
CAFIT		X	X		X	X
ACE			X		X	

Table 4. Three modeling level of detail used. PC keyboard means no schematic entry. Host PC or CAE indicated. Note CAFIT and STAMP were run at two detail levels each.

* If permitted, key items or major test control signals, such as CLOCK or RESET, should be "weighted" to indicate the increases test criticality.

* Such weighting could also indicate high failure items or item complexity, so that a computer box is not equal to a simple junction box or a readout.

* The user must be cautioned that sometimes reliability and test criticality weightings may contradict each other. In this tank model, the CEU was the most reliable (low weight) but the most complex and critical (high weight).

* Mechanical or physical cues available to the technician were utilized in this model. For example, the noise of a gyro or movement of a gun barrel are frequently better information than a Built-in-Test (BIT) light. Such output data can be termed "unobtrusive" testing with no probing required.

* Liberties and imagination must be used to apply non-electronic functions to digital CAD tools. For example in DTA, gates or inverters must represent everything from relays, fuses, wires, and gyros to light bulbs in SCOAP-based TFOM tools.

* Some tools require extra duplicate key entry for names of signals or items. Others have a four-character limit or only print out default computer-assigned alpha-numerics. Thoughtful use of numbers can help. For example, even numbers for inputs and odd numbers for outputs, or different 100 series numbers per object can identify unit and input/output (I/O) status.

2.0 LOGIC MODEL (LOGMOD)

BACKGROUND Ralph DePaul, the president of DETEX Systems Incorporated, first used the dependency tree basis for LOGMOD in the early 1960s. It is by far the most mature TFOM tool around, and was originally intended as a fault isolation maintenance tool. The author has used LOGMOD on two previous board-level TFOM evaluations. LOGMOD has shown considerable revision and enhancement all three times it has been used, which has been about two years apart.

In the 1970's, the primary outputs of the LOGMOD program were a Maintenance Dependency Chart (MDC) which validates fault isolation using a functional half-split test strategy, and an encrypted data structure used to present dynamic test strategy generation. Feedback loops were presented on the MDC, but collapsing the feedback loops was very labor intensive. By 1983, the MDC and the validation of fault isolation were augmented by a new format of the functional half-split test strategy called Logic Test Structure (LTS). The LTS were built in an orthogonal method and presented on 8 1/2 X 11 inch paper. If adjoining pages are placed in their proper positions, they cumulatively visually present the complete strategy for the modeled system. Also in 1983, feedback loops were described in a new "Loop Report", and this new method of collapsing the loops provided a "first look" at inherent ambiguity group sizes. LOGMOD is prevented from running a TFOM report without first eliminating the feedback loop problem. This can be done by redesigning the circuitry, collapsing or breaking the loops.

The WICAT-150 workstation, as shown in Figure 3, has been bundled with the LOGMOD software for several years. Any IBM-compatible PC can also be used by customers in their own facilities. DETEX uses a PRIME 250-II minicomputer with 32 bits, 1 Meg main memory, and 160 MB winchester hard disk drive. Using this much CPU power for compiling and report generation still takes about 1.5 hours to run validations. LOGMOD is written in FORTRAN-77, but an effort is ongoing to rewrite it in the more powerful C language. The Microsoft FORTRAN was inadequate for some intrinsic "AND/OR" function needs.

LOGMOD has one or two-week training classes available, and the software can be utilized by licensing, time sharing, or leasing.

INPUT. LOGMOD input methodology still consists of typing in each output signal "event, the component "item" it comes from, and the input "events" that can drive it. Figure 4 shows some of the LOGMOD dependency input data format used for the tank system. A second list of item and event names must also be typed in. In the age of netlists and CAD entry, this manual writing up and typing of first order dependencies is becoming less desirable than it used to be.

One of the weaknesses in the original LOGMOD was that all events and items were treated equally. In other words, an item spectrum from a resistor or a microprocessor were the same value. DETEX followed user recommendations to enable weight values to be put on important or complex items. In events, CLOCK or RESET signal events should be more test critical than normal data lines. LOGMOD allows the user to assign desired criteria for weighting, which can be reliability, test criticality, or system operation criticality

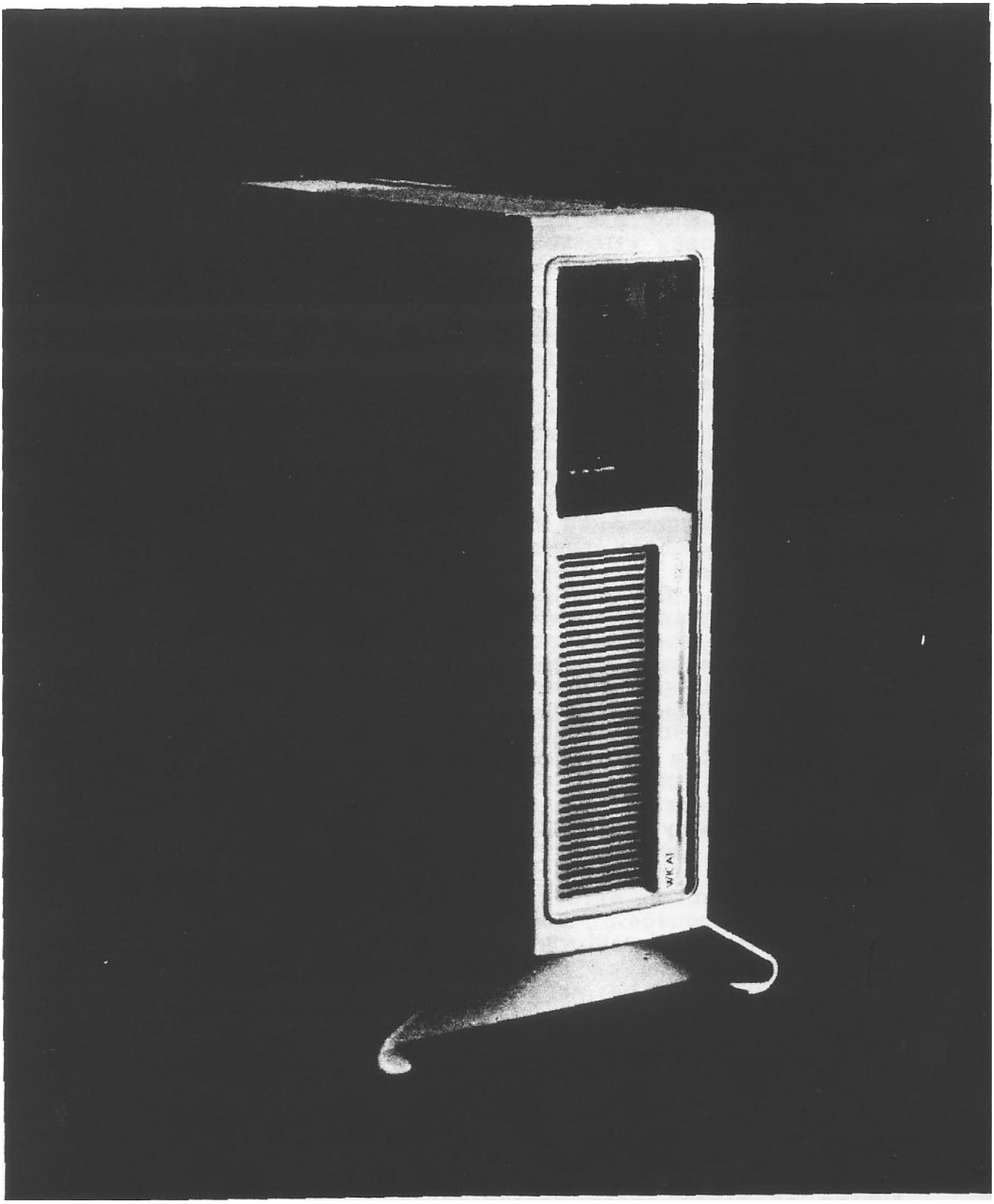

Figure 3. Photograph of WICAT S-1250 computer. Photo Courtesy of WICAT Systems.

```
                1
                2
                3
                4   /THIS DATA BASE IS A DERIVATIVE OF THE M-1 TANK DATA BASE CREATED
                5   /IN 1984. ITS PURPOSE IS TO FORM A BASIS FOR COMPARING DIFFERENT
                6   /TESTABILITY MODELS BY JIM BUSSERT AT NOSC.
                7   /
                8   /
F0-13           9   A1900 I1809 A1228;
               10   A1228;
               11   A1901 I1807 A1900;
               12   A1689 I1809 A1228 A1687;
               13   A1687;
               14   A1690 11817 A1689;
               15   A1700 I1809 A1715 A1757 A1228 A1687 A1059 A1828 A1816 A1057
               16            I1064 A1065 A1764 A1062 A101;
               17   /                    THE FOLLOWING INITIAL EVENTS WERE NOT INITIAL
               18   /                    EVENTS IN THE ORIGINAL MODEL. THEY ARE TREATED
               19   /                    AS SUCH HERE FOR THE PURPOSE OF SIMPLIFICATION.
               20   A1059;
F0-15          21   A1828;
               22   A1816;
F0-13          23   A1057;
               24   A1065;
               25   A1062:
F0-15          26   A1010
F0-13          27   A1701 I1807 A1700;
               28   A1712 I1809 A1715 A1757 A1228 A1687;
               29   41713 I1807 A1712;
               30   41715 I1807 A1714;
               31   A1706 I1809 A1757 A1059 A1764 A1057 I1064 A1065 A1802 A1828
               32         A1816 A1228 A1687 A1715 I1181 A1062:
               33   A1707 11807 A1706;
               34   A1725 I1807 A1724;
               35   A1724 I1809 41733 A1738 A1722;
               36   A1745 11807 A1744;
               37   A1750 I1809 A1745 A1733 A1738 A1715 A1059 A1764 A1057 I1064
               38          I1181 A1065 A1062 A1757 A1228 A1687
F0-15          39   41757 I1840 A1756
               40   A1756 I1804 A1755 A8517
               41   A8517;
               42   A1755 I1840 A1754;
               43   A1754 11809 A1753 A1799 A1793 A1839 A1039;
               44   /                    DEPENDENCIES A1790 THROUGH A1793 ARE FROM
               45   /                    FO-14 AND ARE USED TO ENHANCE THIS MODEL.
F0-14          46   A1790 I1809 I1466 I1464 A1057 I1064 A1065 A1062;
               47   A1791 I1840 A1790;
               48   A1792 I1857 41791 A2021;
               49   A2021;
               50   A1793 11840 41792;
F0-15          51   A1753 I1840 A1752;
               52   A1752 I1804 A2023 A8517;
               53   A2023 11840 A2022;
               54   A2022 I1809 A1057 I1064 A1065 I1464 I1466 A1062;
               55   /                    DEPENDENCIES A1836 THROUGH A1839 ARE FROM
               56   /                    FO-15 AND ARE USED TO ENHANCE THIS MODEL.
               57   A1836 I1809 A1057 I1064 A1065 I1464 I1466 A1062;
               58   A1837 I1849 A1836;
               59   A1838 I1848 A1837 A8519;
               60   A8519;
               61   41839 I1849 A1838;
               62   A1801 I1804 A1755 A8517;
               63   A1802 I1840 A1801
```

Figure 4. LOGMOD dependency data input format. Note events with no dependency, such as A1059, are initial inputs from outside of the system.

OUTPUT All LOGMOD outputs fall within one of seven files. These files are LIST (database input), MANAGEMENT DIAGRAM, LOOPS, MAINTENANCE STRATEGY VALIDATION, TESTABILITY REPORT, ERROR, and SEQUENCE. The loops, error, and sequence files are automatically created and not user selected like the others. LOGMOD has added several new testability output report options recently, and all of the possible options will be overviewed.

1. The list file of the LOGMOD data base inputs (events, items, and names).
2. The Maintenance Dependency Chart (MDC)
3. The Maintenance Strategy Validation

Figure 5 shows a sample page of the Maintainability Validation Report output for the tank system, which will be explained briefly next.

a. Optimal single failure isolation sequence.
   Is based totally on the half-split method of doing diagnostics.
b. Nodes affected by a specific failure.
   Lists an associated event, called A-signal, and lists outof-specification A-events.
c. Diagnosis/repair times for failure.
   Events tested with total diagnostic and repair times.

4. Feedback Loop Report
This includes system/equipment inputs, outputs, item involvement ratios, detailed loop analysis, loop inputs, nodes involved in the loop, node cyclic density, loop outputs, components inside the loop, and aggregate probability of the loop.

5. The Testability Report
This report has very user helpful explanatory paragraphs to clarify the terms, what steps are recommended to improve the design, etc. This extra verbage can be suppressed by the user who just wants the numbers without all of the explanations. Figure 6 shows actual IFIFOM Table and Ambiguity Group output values for the tank system. Figure 7 shows sample Item Involvement Ratios and a BIT/BITE Hierarchy list from the Testability Report.

a. Inherent Fault Isolation Figure of Merit (IFIFOM)
b. Ambiguity Group Recap Table
c. Ambiguity Group Composition Table
d. Item Involvement Ratios
e. MTTI and MTTR
f. Hierarchy For BIT/BITE Candidates
g. System/Equipment Inputs
h. System/Equipment Performance Test Requirements

MAINTAINABILITY VALIDATION

Item: 11809 30 0ccurance(s)

Associated with event A1689 the out-of-specification events art:
A1689 A1690 A1771 A2044 A2561 A2601 A3327 A3332 A2563 A2606
A3285 A3291 A3286 A3292 A3214 A3216 A3218 A3220 A3222 A3224
A3226 A3228 A3230 A3232 A3234 A3236 A3238 A3242 A3244 A3246
A3248 A3250 A3252 A3268 A3271 A3274 A3277 A3280

The event(s) tested are:
A2563 Bad (Terminal event)
A2561 Bad
A2022 Good
A1836 Good
A1790 Good
A2590 Good
A3200 Good
A2669 Good
A3361 Good
A3208 Good
A2668 Good
A2652 Good
A1724 Good
A3363 Good
A2660 Good
A1754 Good
A2639 Good
A1689 Bad
A1687 Good

Total diagnostic time = 57.0 minutes.
Total repair time = 87.0 minutes.

Associated with event A1768 the out-of-specification events are:
A1768 A1769 A3361 A3362 A3363 A3364 A1771 A2044 A2561 A2601
A3327 A3332 A2563 A2606 A3285 A3291 A3286 A3292 A3214 A3216
A3218 A3220 A3222 A3224 A3226 A3228 A3230 A3232 A3234 A3236
A3238 A3242 A3244 A3246 A3248 A3250 A3252 A3268 A3271 A3274
A3277 A3280

The event(s) tested are:

A2563 Bad (Terminal event)
A2561 Bad
A2022 Good
A1836 Good
A1790 Good
A2590 Good
A3200 Good
A2669 Good
A3361 Bad
A1768 Bad
A1059 Good

Total diagnostic time = 33.0 minutes.

Figure 5. LOGMOD Maintainability Validation data. Note: "terminal events" are outputs to the external world from the system in LOGMOD terms.

Inherent Fault Isolation Figure Of Merit Table

Col. 1 Ambiguity Group Size	Col. 2 Worst Case Indicator	Col. 3 Best Case Indicator	Col. 4 Delta	Col. 5 Cumulative Range (Lo - Hi)
1	38.182	80.769	42.587	TO 80.769
2	3.636	3.846	.210	41.818 T 84.615
3	10.909	7.692	3.217	TO 92.308
4	7.273	3.846	3.427	60.000 T 96.154
22	40.000	3.846	36.154	TO 100.000

Ambiguity Group Recap

Ambiguity Group Size	Occurance(s)	Cumulative Occ(s).
1	149	149
2	1	150
3	2	152
4	3	155
22	6	161

Figure 6. LOGMOD IFIFOM and Ambiguity Group data.

Item Involvement Ratio(s)

Item(s)		Failure Rate	Leverage	Weighted Leverage
11064	TNB CB 13	.150e+03	3.987	.598e+03
F1181	TNB CB 30	.150e+03	1.993	.299e+03
I1464	TNB CB 31	.150e+03	1.329	.199e+03
11466	TNB CB 32	.150e+03	.997	.150e+03
11804	GPS	.180e+03	5.648	.102e+04
I1806	GCH HANDLE	,100e+03	2.658	.266e+03
11807	1W200 CABLE	.150e+03	7.973	.120e+04
11808	CCH HANDLE	.100e+03	2.658	.266e+03
11809	TNB	.235e+03	9.967	.234e+04
I1813	GTD	.150e+03	4.651	.698e+03
11814	LOS	.150e+03	3.987	.598e+03
I1815	1W202 CABLE	.230e+03	7.973	.183e+04
I1816	1W204 CABLE	.100e+03	1.993	.199e+03
11817	1W201 CABLE	.230e+03	4.319	.993e+03
11821	CEU	.280e+03	3.987	.112e+04
I1840	1W104 CABLE	,150e+03	4.651	.698e+03
11848	LDR'S PANEL	.800e+02	.332	page 25

Node	Title	Count	Leverage	Weighted Leverage
A2666	LOS EL DRIFT ADJ	190	2.641	.822e+01
A2653	LOS EL HAND SIG	173	2.405	.730e+01
A1725	GTD AZ ST GYRO	149	2.071	.624e+01
A2561	EL SERVO INPUT	147	2.044	.669e+01
A1802	TNB EMER SW IN	129	1.793	.531e+01
A3292	CCP CEU CLK *IN*	116	1.613	.467e+01
A1757	TNB NORM MODE IN	102	1.418	.416e+01
A1745	TNB DC AZ HAN HI	100	1.390	.407e+01
A3364	*CEU* LDRS RDY IN	94	1.307	.385e+01
A3291	*CEU CCP* CLK OUT	90	1.251	.324e+01
A3332	CEU CANT SIC IN	89	1.237	.318e+01
A3211	CEU CCP PWR ON	88	1.223	.346e+01
A2661	CEU *EL ST* GYRO	84	1.168	.321e+01
A1332	CEU DCT ERROR RT	78	1.084	.292e+01
A2570	GTD GUN GYRO IN	72	1.001	.256e+01
A2552	GTD TURRET GYRO	70	.973	.250e+01
A1690	CEU PALM SWS IN	68	.945	.245e+01
A3327	CEU CCP SW IN	67	.931	.240e+01
A2044	CEU WIND SENS IN	45	.626	.163e+01
A1801	CPS EMER SW OUT	27	.375	.115e+01
A1755	CPS F.C. SW IN	26	.361	.111e+01
A3286	CCP CEU DATA IN	26	.361	.143e+01
A1754	TNB F.C. OUT	25	.348	.106e+01
A1839	TNB GPS MAN IN	24	.334	.102e+01
A2652	TNB EL SIC OUT	24	.334	.106e+01

Figure 7. LOGMOD Item Involvement (top half) and Built-in-Test (BIT) hierarchical candidates (lower half) with leverage .

6. Logic Test Structure (LTS) LTS-30
   The LOGMOD data base two LTS-30 tree chart outputs are:

   a. Checklist LTS
      The checklist LTS is a step-by-step fault isolation sequence that can handle multiple fault modes. The checklist LTS is constructed to take advantage of good-pass and bad-fail test information previously performed. This is to eliminate unnecessary performance of already proven good-pass tests.
   b. Independent LTS
      The independent LTS differs from the checklist LTS by not considering any previous test information The independent LTS starts with a known observed failure and then backs up to the initial problem. Each system output is considered in a unique test structure, and the split-half strategy is applied.

COMMENTS The variety of outputs from the LOGMOD are certain to meet testability and other field maintenance or logistic needs of users. The feedback loop data and detail from LOGMOD is a strong point. It had the first BITE hierarchy listing, which is still one of the more useful DFT outouts. Several years ago DETEX and a large aerospace firm developed and field tested a Portable Maintenance Aid (PMA) for flightline fault isolation, and LOGMOD has been used for logistic and repair analysis.

2.1 SYSTEM TESTABILITY ANALYSIS MAINTENANCE PROGRAM (STAMP)

BACKGROUND The ARINC Research Corporation office in Annapolis, MD developed the STAMP logic model in the early 1980s. It originally was resident on an Apple II PC and included very user friendly test analysis printouts. Around 1985, the program was rewritten for larger tasks and Figure 8 shows the Hewlett-Packard HP-100 A900 mini-computer now used by STAMP. The user friendly printouts have not been developed for the HP STAMP as of late 1987. ARINC policy is that STAMP will be used only by their own employees and primarily for government contract work, with some commercial applications possible. Therefore there is no real need for STAMP user books or explanatory printouts. However a user's guide and series of technical notes have been written. ARINC delivers a full report contract document complete with analysis and recommendations. If a company needs a TFOM report on a government system, and does not want to train in-house people to do this, STAMP would be a good third party contract option.

This is one of the two TFOMs that the author ran at two model input levels. The author anticipated just running the dependency logic input on STAMP, but ARINC recently has added a schematic CAD input capability from a PC using Schema schematic capture software. STAMP now has three input capabilities, which are PC CAD, PC word processor, or normal STAMP utilities. ARINC claims that nearly 50 percent of the current STAMP program is new within the last year or so. Despite the obvious input similarities, LOGMOD and STAMP have many output and operational differences. For example, STAMP does not require that feedback loops be broken prior to running, which LOGMOD does, and STAMP can group test in user-defined areas such as cabinets, electrical, or mechanical. The 24 Testability Measures that are generated by STAMP will be shown in the STAMP output section.

Like DETEX with LOGMOD, ARINC has used STAMP for considerable Integrated Logistic Support (ILS), Failure Mode Effects Analysis (FMEA), and troubleshooting guidance contracts. For example, STAMP performs interactive fault isolation on a lap top DATAGEN briefcase tester. This report will stress the TFOM and DFT aspects of STAMP as within the scope of the evaluation.

INPUT (HP-1000) A good overview of STAMP functions can be gained by examining the STAMP Function Menu on the upper half of Figure 9. Menu items two and three are to create a new model and edit it. There were considerable input difficulties and reentering of the first order dependency data several times. This was because of insufficient time available to become acquainted with the user interface prior to actual use of STAMP. Figure 10 illustrates the input dependency format for STAMP. A first order dependency input code can be entered into either LOGMOD or STAMP with minor format differences. For example, LOGMOD Items and Events are termed Components and Tests in STAMP.

The sequence of entering STAMP data is first to specify the number of tests, components, testable inputs, and testable outputs; next create labels for each element; and then enter the system functional dependencies. To avoid default labels, user-assigned labels, with a maximum of 16 spaces, should be entered prior to the dependency. It was desired to enter reliability and test criticality weights into the STAMP model analysis. The STAMP component weight is termed "failure frequency", however STAMP allows redefining it per the users desire. STAMP test weight values were assigned to some critical signal areas such as the TNB and CEU.

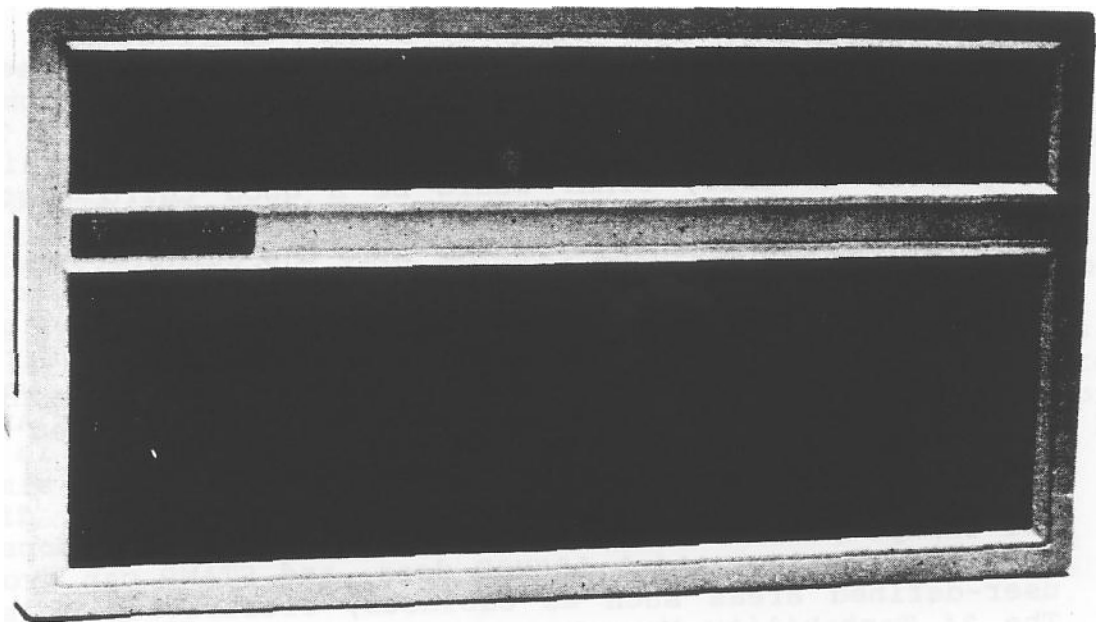

Figure 8. Hewlett-Packard 1000 A900 minicomputer. Photo courtesy of Hewlett-Packard

```
Current File: TNEXP                                                    STAMP Version 2.3
Description:    Example from Tecnotes 321.1 and 331.1
Date: 9-18-86                Testable Elements:        20 Fault Isolation Elements:        26

 STAMP Function Menu
 ------------------------------------------------------------------------------------------------------------------------------------
 0.    Exit to STAMP System Menu
 1.    Select current file name
 2.    Edit current file
 3.    Create new file
 4.    Compile dependency model (knowledge-base)
 5.    Compute testability
 6.    Generate testability report
 7.    Generate fault tree
 8.    List file to screen/printer (see NEWS in SM)
 9.    Conduct interactive fault isolation (Inferential)
10.    Conduct interactive fault isolation (Evidential)

------------------------------------------------------------------------------------------------------------------------------------
Enter an option number >
```

```
Current File: TNEXP                                                    STAMP Version 2.3
Description:  Example from Tecnotes 321.1 and 371.1
Date: 9-16-86                Testable Elements:        20 Fault Isolation Elements:        26

Test Report Selection ------------------------- Press RETURN when complete
------------------------------------------------------------------------------------------------------------------------------------
    *1.  BASIC TESTABILITY MEASURES                    11.   Full Subsignature List
     2.  Extended Testability Measures                 12.   No Input Subsignature List
     3.  Conditional Testability Measures              13.   FALSE FAILURE TABLE
    *4.  GROUP TESTABILITY MEASURES                    14.   False Failure by Group
     5.  List of Component Groups                      15.   List of Test Groups
    *6.  COMPONENT AMBIGUITY TABLE                     16.   All of the above
     7.  Excess Test List                              17.   Single Failure FMEA
     8.  Excess Test Analysis                          18.   Multiple Failure FMEA
     9.  Feedback List                                 19.   Failure Frequency Report
    10.  Test Redundancy List                          20.   Higher Order Dep. Report

                                  21. Exit no report
                  *******Selecting a Number Will Toggle the Selection********
          ------------------------------------------------------------------------------------------------------------------------------

Enter a report item number>
```

Figure 9. STAMP Function menu (top half) and Testability Report menu (lower half).

INPUT (PC UTILITIES) It took only one morning to enter the block diagram and download it to STAMP on the HP-1000. When using the Schema input to PC, mouse driven menus are used. In addition to the traditional STAMP "tests" and "components", Tests, Testable Inputs (TI) and Untestable Inputs (UT) are specified by labeling the nets of the block diagram. Schema rules will not allow two labels on one net, so a pseudo "box" may need to be drawn. An example of this is in the PC block schematic Figure 11, with the Component box D1, which ties multiple lines together as a pseudo AND gate. Due to hierarchical limitations, the five CBs in the TNB box were represented as TNB1 through TNB5, with connections as the CBs are inside of the TNB. The drawing model was made in four steps which are Draw, Object, Component, and Designator. ARINC had a library of simple block components, such as a one-input/3-output box. The Schema post-processor lists labels, components, and error checks. An ARINC developed utility written in dBASE III is then used to convert the Schema wire list into a dependency list and downloads it into STAMP. This PC to HP conversion program (STAMPREP) has a limitation of no more than 20 inputs to the portion of the system modeled. However, multiple diagrams may be merged into a simple model using STAMPREP or another HP utility called COMBINE.

OUTPUT (HP-1000) FULL-NODE MODEL A portion of the full dependency STAMP test report with 24 test measures, is shown in Figure 12. As the Testability Measure names indicate, several of the outputs are similar in meaning and purpose. A detailed description of the meaning for all 24 outputs will not be attempted, but some of the main output results will be noted, as explained by ARINC personnel.

The value of Test Leverage (TL) should lie between the values of Theoretical Minimum (Theomin) and Theoretical Maximum (Theomax). Since TL, which measures the test set efficiency, is greater than Theomax, this indicates that there are too many Tests. The Excess Test Measure (XM) value of 80 percent, which is the percentage of tests that are candidates for elimination, also says the same thing. This brings up the same modeling problem common to LOGMOD. The algorithm assumes that all nodes are observable, and so the complete system model is seen as all output test points. Naturally this gave a high figure for Isolation Level (IL). The Feedback Modified Test Leverage (FMTL) value is high, and elimination of the feedback loops would not bring it to a desired level. A high value of Hidden Failure Measure (HFM) indicates problems with multiple failures, but a low Percent HFM (PHFM) indicates there will be few multiple failures. A large Input Modified False Failure Measure (IMFFM) indicates a high probability of false failures. The related Input Modified HFM (IMHFM) is a high 76 percent, which correlates to the high External Dependency (EXDEP) external control of 73 percent. The False Alarm Tolerance (FAT) value is good however. The Dependency (DEP) and Test Interdependency (TIDEP) values should be within 20 percent of each other, and are in this model. NON-TEST POINT NODES DELETED To try and correct the 360 testable node model, and make the Test Report more realistic, all internal dependency data was omitted. This left only those tests tying to 90 external inputs and 30 external outputs tests. This resulted in far too much being deleted. A High Order Dependency Report, option 20 on the Test Menu, was run as a QA check, prior to more detailed test report options. The new test report said there were not any feedback loops in the circuit, which is a main testability problem with the tank FCS design. ARINC then advised that deletions should have been made according to a Recommended Excess Test List. This is a special "excess test" report option for optimum test reduction. This "fix" would have required adding 190 tests and dependencies back into the STAMP model again. Unfortunately the main file was accidentally erased and there was not time enough to "recreate" it all again prior to making more "what if" changes.

```
File Name: JIMDP
File Description: File with dependencies
File Date: 3/18/87

Test     9    Label  = (TE9)     A1315
Dependencies:
(TE197)    A2649           (TE210)    A2664           (TE327)    A8150
(TE328)    A8151           (7E329)    A8152           (TE330)    A8153
(TE331)    A8154           (7E332)    A8155           (TE333)    A8156
(TE334)    A8157           (TE335)    A8158           (TE336)    A8159
(TE337)    A8160           (TE338)    A8161           (CP11)     I1814

Test     10   Label  = (TE10)    A1316
Dependencies:
(TE9)      A1315           (CP12)     I1815

Test     11   Label  = (TE11)    A1317
Dependencies:
(TE13)     A1319           (CP12)     I1815

Test     12   Label  = (TE12)    A1318
Dependencies:
(TE11)     A13)7           (CP9)      I1809

Test     13   Label  = (TE13)    A1319
Dependencies:                   (CP11)     I1814
(TE191)    A2640

Test     14   Label  = (TE14)    A1329
Dependencies:
(TE10)     A1316           (CP9)      I1809

Test     15   Label  = (TE15)    A1330
Dependencies:
(TE14)     A1329           (CP12)     I1815

Test     16   Label  = (TEIG)    A1332
Dependencies:                   (CP12)     I1815
(TE12)     A1318

Test     20   Label  = (TE20)    A1689
Dependencies:
(TI8)      AI228           (TI11)     A)687           (CP9)      I1809

Test     21   Label  = (TE21)    A)690
Dependencies:
(TE20)     A1689           (CP14)     I1817
Test     22   Label  = (TE22)    A1700

Dependencies:
(TE29)     A1715           (TE51)     A1757           (TE53)     A1764
(TE77)     A1828           (TI1)      A)057           (712)      A1059
(TI3)      A1062           (TI4)      A1065           (7I8)      AI228
(TI11)     A1687           (TI14)     A1816           (CP1)      I1064
(CP9)      I1809

Test     23   Label  = (TE23)    A1701
Dependencies:
(TE22)     A)700           (CP7)      I1807
```

Figure 10. STAMP input dependency format. This is typed in similar to LOGMOD.

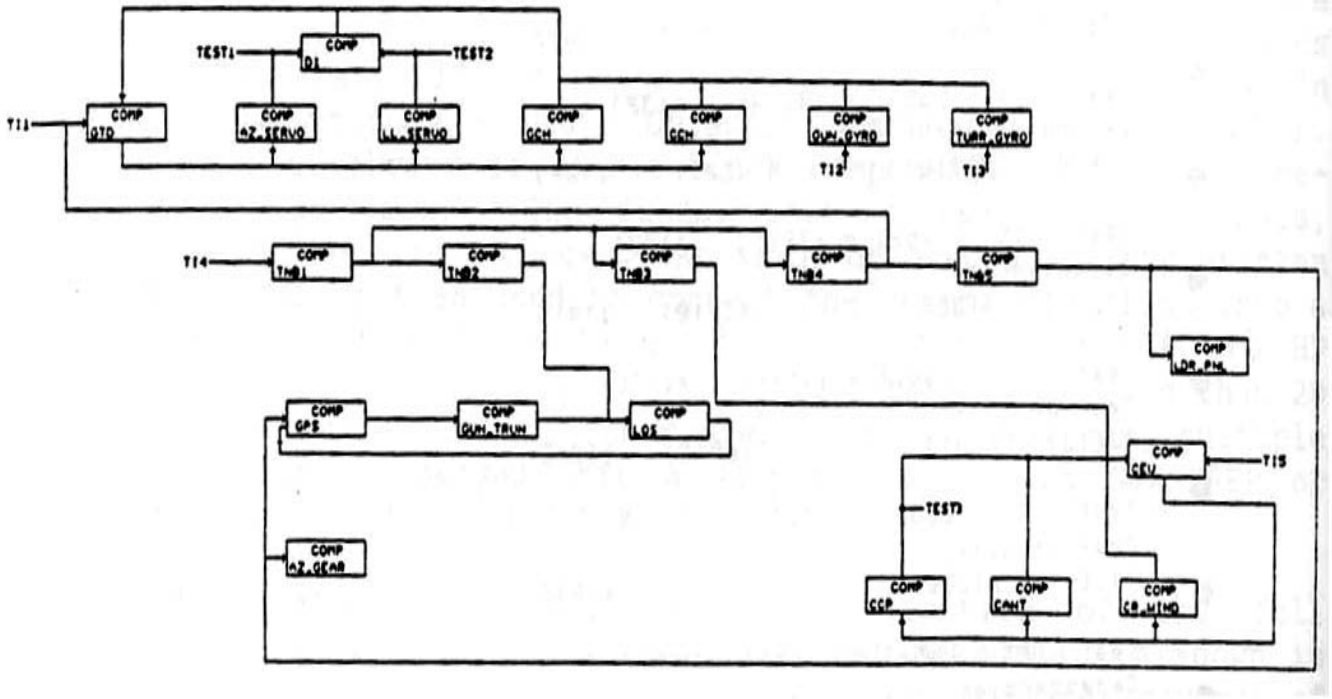

Figure 11. STAMP PC block diagram. This was done on the PC using Schema schematic software.

```
MODULE 2                2:13 PM WED., 18       MAR., 1987
  Testability Measures                    JIMDP

  .9431     = IL              Isolation Level

 1.0000     = FMIL            Feedback-modified Isolation Level

 2.8699 = TL                  Test Leverage

 1.6829     = NRTL            Nonredundant Test Leverage

 2.1466 = FMTL                Feedback-modified Test Leverage

  .5864 = TU                  Test Uniqueness

  .4136     = TR              Test Redundancy

  .2975 = TFBD                Test Feedback Dominance

  .0650 = CFBD                Component Feedback Dominance

 0.0000     = NDP             Nondetection Percent

  .9174     = HFM             Hidden Failures Measure

  .7667 = IMHFM               Input-modified Hidden Failures Measure

   .0131    = PHFM            Percent Hidden Failures Measure

   .0722 = IMPHFM            Input-modified Percent Hidden Failures Measure False

   .0083 = FFM               Failure Measure

   .0333 = IMFFM             Input-modified False Failure Measure

  .3601     = DEP             Dependency

  .3463 = TIDEP               Test Interdependency

  .3997 = TDEP                Test Dependency

  .3473 = FAT                 False Alarm Tolerance

  ,0564 = THEOMINTL           Theoretical Minimum Test Leverage

  .9919 = THEOMAXTL           Theoretical Maximum Test Leverage

  .7398 = EXDP                External Dependency

  .7960 = XM                  Excess Test Measure (Includes Redundants)
```

Figure 12. A portion of STAMP the full-node HP testability report. The meanings of key values are described in the text.

OUTPUT (PC UTILITY) The PC to HP conversion program STAMPREP, consists of FASTPREP, FASTADD, TESTMAKER, and FILEPREP utilities. A look at Figure 11 shows two obvious feedback loops at CCP to CEU and LOS to GPS. Only the first is noted by the STAMP and the second is not observable. The Schema block level STAMP outputs are shown in Figure 13.

COMMENTS The addition of CAD schematic inputs to STAMP are a major improvement. It still seems that 24 esoteric numbers are rather user hostile and possibly overkill. However there is much more to the report, as indicated in the lower half of Figure 9. STAMP is aimed primarily at the military contract market, though some commercial work is done. If a company desires an outside party to do the work and deliver a polished report, STAMP would be the way to go. It would seem that there could be contractors that would want to have their own engineers use STAMP. This possibility would require a change in ARINC management philosophy in the future. There were three items that stood out as neat things that I would hope to see emulated by other TFOMs. One was the entry for skill level of the technician. Next was the test penalty weight in time or cost. Finally the capability for the user to group tests functionally. The application of STAMP to fault isolation uses is beyond the scope of this study, but worth noting here.

MODULE 2 1:05 PM THU., 19 MAR , 1987
Testability Measures JIMC

.2963	= IL	Isolation Level
.5000	= FMIL	Feedback-modified Isolation Level
.2963	= TL	Test Leverage
.2593	= NRTL	Nonredundant Test Leverage
.4375	= FMTL .	Feedback-modified Test Leverage
.8750	= TU	Test Uniqueness
.1250	= TR	Test Redundancy
.2750	= T FED	Test Feedback Dominance
.4815	= CFBD	Component Feedback Dominance
.2963	= NDP	Nondetection Percent
.2000	= HFM	Hidden Failures Measure
0 0000	= IMHFM	Input-modified Hidden Failures Measure
0.0080	= FHFM	Percent Hidden Failures Measure
0.0000	= IMPHFM	Input-modified Percent Hidden Failures Measure
0.0000	= FFM	False Failure Measure
0 0000	= IMFFM	Input-modified False Failure Measure
0.1643	= DEP	Dependency
0.2031	= TIDEP	Test Interdependency
0.1528	= TDEP	Test Dependency
0.2321	= FAT	False Alarm Tolerance
0,1761	= THEOMINTL	Theoretical Minimum Test Leverage
0 9630	= THEOMAXTL	Theoretical Maximum Test Leverage
0.1852	= EXDP	External Dependency
0.1250	= XM	Excess Test Measure (Includes Redundants)

Figure 13. A portion of the STAMP PC block Testability Report. The difference in the values are interesting to indicate the trade-off of detailed versus high level block modeling.

2.2 ADVANCED SYSTEM TESTABILITY EVALUATION PROGRAM (ASTEP)

BACKGROUND BITE, Incorporated in Manassas, VA developed a new ASTEP TFOM tool to be available in early 1988. The ASTEP was run on an IBM-AT, as illustrated in Figure 14, however any IBM-compatible PC can be used, and floppy disk programs and user manuals will be available. The ASTEP Data Entry Program (ADEP) uses a commercially available Ashton-Tate DB-III. The ASTEP itself uses a compiled DB-III that is not required by users.

The entire process and philosophy of ASTEP is quite different from the usual logic model dependency family of TFOMs. ASTEP test outputs are test performance measures found in most test requirement specifications. Being an ex-test programmer, the author felt at ease with this TFOM approach. This TFOM has its baggage of unique terms summarized in Table 5, that the ASTEP user must become familiar with. They are keyed as to their area of application, which can be modeling, input data, or output values of ASTEP.

A test performance prediction is made by performing a test coverage estimate (TCE) over a set or collection of hardware faults called a Fault Quantum (FQ) which has a failure rate assigned to it. A FQ, which is one or more devices or components, relates the physical hardware to the functional structure. It can be hierarchical and can include as many levels of details as is desired. The TCE data entry consists of the Test ID (TESTID), the Fault Quantum ID (FQID), and the Probability of Fault Detection (PFD) of the test on the FQ. The test coverage estimate can be derived from either manually generated test coverage estimates or from the output of other TA tools such as fault simulators. The usual 31 system items of boxes and cables had to be increased to 45 due to the added component hardware paths through boxes in the test assessment. Figure shows the actual tank FCS system path test assessment sensitized paths drawn for this ASTEP model.

INPUT ASTEP input is done with the ADEP DBIII program Browse command. The primary input screens are shown in Figures 16(a) and 16(b). The hierarchical FQ levels of the model are shown by the three left-hand vertical columns of the data entry screen.

One or more components must be listed for each FQ identifier. The components are usually identified by their reference designators. Hardware levels (unit, module or IC) are indicated by spaces within the reference designator field. Four maintenance environments are possible within one ASTEP database. A default replacement hardware level is specified for each maintenance environment, unless an exception is specified by the value in the columns labeled RL#. This ability to indicate level of replacement is not available in most other TFOM tools.

In the example shown, the LOGMOD item numbers were entered into the REFDES field for continuity across various tools. Next is the failure rate (FR) of the component. The FBID input data is used to provide another modeling hierarchy such as block

3.0 SUMMARY

The intent of the modeling effort was not to arrive at a preferred TFOM modeling method or even to determine which parameter(s) are the ideal TFOM value to meet MILSTD-2165 criteria. As a matter of fact, these eight tools used a total of 62 test report outputs. These were not all unique, and six values appeared more than once.

The purpose of this report was to show contractors and government employees what tools are available that can do system level TFOM analysis. Inputs, outputs, advantages and limitations of each tool are described. Based upon this data, a future user can pick the optimum TFOM tool for a particular system application. The modeling hints and recommendations in section 1 should be valuable also. It is possible that a user may want to use more than one TFOM during a system design process. One TFOM could provide a quick and dirty high level evaluation, and another TFOM more detailed specific redesign level analysis.

The number of new TFOM tools coming on-line this year is encouraging and all are CAD or PC-based. The choice of available TFOM tools have doubled in the last two years, and many include useful TPS, ILS, and fault isolation features in addition to TFOM outputs. Table 8 summarizes the features of the computers used by the vendors in this report.

Some of the logic model tools require deletion of sections of the system that are not output observable, since all nodes are assumed testable. Methods of assigning severe test penalties to non-observable nodes, or comparing a total model and testable area databases, seem better approaches.

3.1 **RECOMMENDATIONS**

* Mechanical CAD workstations need netlist or interconnect outputs similar to the electronic CAD outputs, to enable testability analysis of both technologies.

* Although desk top PCs with microcomputers are fine for chip or board level work, a mini-computer CPU is needed for compiling system level simulations or analysis.

* CAD vendors need to combine electronic and non-electronic software into a mixed system for true system applications.

* Only TFOM tools that use "functional" rather than "gate level" are applicable to system test assessment.

* The Daisy DTA is excellent for ASIC and digital module testability analysis, but not intended for high level system use.

* TFOM static condition models cannot handle multi-mode analysis with different configuration changes, each with possibly different "testability". Non-static TFOMs are needed. CAFIT comes closest to this need among these tools.

	LOGMOD	STAMP	A-STEP	IN-ATE	DTA	CAFIT	ACE	2165
IOC	1975	1981	1985	1984	1982	1987	1988	1985
HOST	WICAT	HP1000	IBM-AT	APPLE	DAISY	IBM-AT MENTOR	SUN	NA
OPERAT SYSTEM	UNIX	RTE-A	PC-DOS	APPLE	DNIX	PC-DOS UNIX	UNIX	NA
BUS	MULTI BUS	NO	AT BUS	NU BUS	MULTI BUS	AT MULTI	VME	NA
DATA XFER	ETHER NET	NO	?	ETHER NET	ETHER NET	MENTOR DOMAIN	NO	NA
SCHEM INPUT	NO	YES	NO	YES	YES	YES	YES	NA
SCHEM SW	NO	SCHEMA	NO	EDIF	DED-2	CASE MENTOR	CUSTOM	NA
LANGUAGE	FORT-RAN	FORT-RAN	DB-III	C	PASCAL	FORT-RAN	FORT-RAN	NA
METHOD	DEPEND-ENCIES	DEPEND-ENCIES	FAULT MAP	AI LOGIC	SCOAP	MOD DMA	CONN-ECT	CHK LIST
USER MANUAL	YES	YES	YES	YES	YES	YES	DRAFT	NA
MENUS	NO	YES	YES	YES	YES	YES	YES	NA
TRAIN CLASSES	YES	NO	NO	YES	YES	NO	NOT YET	NA
MTBF INPUT	YES	YES	YES	YES	NO	YES	YES	NA
FAULT ISOLAT	YES	YES	YES	YES	NO	NO	YES	NA
GOVT ONLY	NO	NO	NO	NO	NO	YES	YES	NO

Table 8. TFOM computer feature summary.

* TFOM tools need to handle non-digital devices and systems in a functional way. A "box" is too simplistic, and "gate" representations are too memory hungry and laborious.

* TFOM modeling of software and hardware, especially in BIT applications, need to be addressed since "hardware connect" TFOM is incorrect with microcode, Built-in Logic Block Observation (BILBO), scan set designs, and so forth.

* The trend to weighted reliability input to TFOMs is good, but a user designated weighting is needed for "test criticality" or mission critical, etc.

* TFOM test data outputs should match Test Program Set (TPS) and system contractual data needed. For example, several algorithms include fault isolation (FI) trees, but no maximum FI time output.

* Many logic model TFOMs assume a test point at every modeled node, which requires deletion of portions of the model. Methods to separate access nodes from other non-accessible nodes are needed, such as test costs for example.

* TFOM reports should printout user assigned names to tests and devices and have schematic and printout names match-up.

* Analog SPICE models need a TFOM method to allow integration of SPICE with SCOAP for mixed technology or hybrid system TFOMs.

* The limited number of label digits of some TFOMs, require abbreviations of names that are not understandable by the user. Larger character fields are needed on these.

3.2 CONTACT POINTS

To facilitate direct liaison to the TFOM sources or vendors for questions or possible applications, the following list is provided.

LOGMOD: DETEX Systems, Inc.
Attn: Ralph DePaul
17871 Santiago Blvd, Suite
221 Villa Park, CA 92667
(714) 637-9325

STAMP: ARINC Research Corp.
Attn: Dr. Randy Simpson
2551 Riva Road
Annapolis, MD
21401
(301) 266-4066