

2

ATION PAGE

Form Approved
OMB No. 0704-0188

AD-A233 260

average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Avenue, Washington, DC 20540, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. .		3. Report Type and Dates Covered. Proceedings	
4. Title and Subtitle. A Hierarchical Artificial Intelligence Maintenance Advisor		5. Funding Numbers. Program Element No. 256020H Project No. 00101 Task No. 101 Accession No. DN257009	
6. Author(s). Joseph A. Molnar and George J. Moss, Jr.		8. Performing Organization Report Number. PR 90:022:252	
7. Performing Organization Name(s) and Address(es). Naval Oceanographic and Atmospheric Research Laboratory Ocean Acoustics and Technology Directorate Stennis Space Center, MS 39529-5004		10. Sponsoring/Monitoring Agency Report Number. PR 90:022:252	
9. Sponsoring/Monitoring Agency Name(s) and Address(es). Naval Sea Systems Command Washington, DC 20362-5101		11. Supplementary Notes. ADPA	
12a. Distribution/Availability Statement. Approved for public release; distribution is unlimited.		12b. Distribution Code. DTIC ELECTE APR 02 1991 S G D	
13. Abstract (Maximum 200 words). The AN/SQS-538 sonar system is used on Navy Ticonderoga class cruisers. Signal detection and processing functions are performed in the system with a mixture of digital and analog modules. System monitoring functions are provided to warn operators of performance degradations and to isolate faults. Monitoring is not equally effective at providing diagnostic information for all subsystems, e.g. the Coded Pulse Processor. A Research and Development project was conducted to develop an artificial intelligence software package to assist technicians in maintenance. The AI software will be resident on a centrally located computer and accessible via a serially connected touch screen terminal. Developments were made to integrate capabilities found in FIS and Rule-master software, and to provide enhancements. An architecture was demonstrated consisting of a hierarchy with two layers: Functional and Isolation. The circuit topology of the coded Pulse Processor provided the basis for knowledge acquisition. The hierarchy significantly reduces the processing time required for the large knowledge base. The hierarchy replicates the system strategy for fault isolation. System monitoring state information is entered by a technician into the Functional layer of the hierarchy. The Functional layer determines if the Coded Pulse Processor is performing improperly and identifies the functional abnormality. Control is then directed to the Isolation layer. The Isolation layer directs the technician to perform the most efficient, detailed troubleshooting tests. Tests are recommended to the technician based upon results of previous tests and the cost required to perform additional tests. Based upon the relative probability information accumulated in testing, recommendations for replacements are made. Multiple fault replacements can be performed, or system testing can be redirected to another functional area.			
14. Subject Terms. (U) Artificial Intelligence; (U) Maintenance		15. Number of Pages. 10	
		16. Price Code.	
17. Security Classification of Report. Unclassified	18. Security Classification of This Page. Unclassified	19. Security Classification of Abstract. Unclassified	20. Limitation of Abstract. SAR

DTIC FILE COPY

PROCEEDINGS

AMERICAN DEFENSE PREPAREDNESS ASSOCIATION

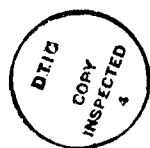
DEPUTY ASSISTANT SECRETARY of DEFENSE (LOGISTICS)

DIRECTOR for LOGISTICS (14)

DCS LOGISTICS—MILITARY SERVICES



Artificial Intelligence Applications for Military Logistics



27 - 30 March 1990

Fort Magruder Inn and Conference Center
Williamsburg, Virginia

91 3 25 039

Availability Codes	
Dist	Avail and/or Special
A1	20

A HIERARCHICAL ARTIFICIAL INTELLIGENCE MAINTENANCE ADVISOR

Joseph A. Molnar*
Naval Research Laboratory
Washington, DC 20375
(202) 767-0327

George J. Moss, Jr.
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529
(703)602-3184

The AN/SQS-53B sonar system is used on Navy Ticonderoga class cruisers. Signal detection and processing functions are performed in the system with a mixture of digital and analog modules. System monitoring functions are provided to warn operators of performance degradations and to isolate faults. Monitoring is not equally effective at providing diagnostic information for all subsystems, e.g. the Coded Pulse Processor.

A Research and Development project was conducted to develop an artificial intelligence software package to assist technicians in maintenance. The AI software will be resident on a centrally located computer and accessible via a serially connected touch screen terminal. Developments were made to integrate capabilities found in FIS and Rulemaster software, and to provide enhancements. An architecture was demonstrated consisting of a hierarchy with two layers: Functional and Isolation. The circuit topology of the Coded Pulse Processor provided the basis for knowledge acquisition. The hierarchy significantly reduces the processing time required for the large knowledge base.

The hierarchy replicates the system strategy for fault isolation. System monitoring state information is entered by a technician into the Functional layer of the hierarchy. The Functional layer determines if the Coded Pulse Processor is performing improperly and identifies the functional abnormality. Control is then directed to the Isolation layer. The Isolation layer directs the technician to perform the most efficient, detailed troubleshooting tests. Tests are recommended to the technician based upon results of previous tests and the cost required to perform additional tests. Based upon the relative probability information accumulated in testing, recommendations for replacements are made. Multiple fault replacements can be performed, or system testing can be redirected to another functional area. Functional layer knowledge must be tailored to the system. Knowledge bases are required for each functional area of the Isolation layer. System architecture and results are discussed.

A HIERARCHICAL ARTIFICIAL INTELLIGENCE MAINTENANCE ADVISOR

Joseph A. Molnar*
Naval Research Laboratory
Washington, DC 20375
(202) 767-0327

George J. Moss, Jr.
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529
(703)602-3184

INTRODUCTION

The Navy's Ticonderoga class cruiser fields an advanced assortment of electronic systems. This class of ship was specifically designed as the platform for the the Aegis system. It has the latest developments in warfare systems for surface combatants. Included (beginning with the CG-56) in the list of electronic systems is the active sonar system, AN/SQS-53B. The AN/SQS-53B uses twelve processing channels, four left, four center and four right. Transmission and reception of acoustic signals is performed with mostly analog circuitry, which is predominantly used in the system front end. After detection, signals are processed by the Coded Pulse (CP) Processor, where an analog to digital conversion is performed. Signals beyond this point in the system (or downstream in the circuit path) are processed digitally.

The AN/SQS-53B has the capability to monitor the system's internal performance at the functional level. Initial information of system performance is obtained from the display console, where each of the twelve channels has a display track on a single CRT. The information presented on the CRT relates the return signal to moving objects, stationary obstacles, clutter, noise, etc. Either, overabundance or absence of clutter and noise, may provide the first indication that the system is malfunctioning. A built-in self test subsystem, Performance Monitoring Fault Locator (PMFL), provides operational test information. There are also test generators which present off-line, or mode specific test vectors to indicate system performance. The functional tests require a minimal level of effort by the operator, i.e., switch selection. Other tests are available as a secondary level of performance monitoring. These are tests that require cabinets to be opened so that indicator light responses may be noted. All are performance tests since they assess the normal functionality of the system. Each of the performance tests is linked to functions executed by the system.

Beyond performance tests, diagnostic tests are used to isolate the cause of an abnormality in a monitored function. Diagnostic tests are performed by trained technicians, who monitor test responses with general purpose test equipment

(GPETE) connected at system testpoints. Troubleshooting is based upon training, experience and technical manual information.

An Artificial Intelligence (AI) system, named Technician's Assister System (TAS), was developed to assist technicians in the maintenance process. A hierarchy was developed for TAS which utilizes the sonar system's internal monitoring capability to direct a technician to the proper functional area. Further assistance is supplied to the technician by the suggestion of diagnostic tests. The intended result leads the technician to a suggestion for card (module) replacement. Testing is directed along a path which is most efficient as determined by resultant information or relative test execution cost. Replacement is based upon probabilistic reasoning.

Specifically, the TAS is a proof-of-concept model to address troubleshooting of the CP Processor. The CP Processor was selected because: (1) digital and analog circuits are both present; (2) there are gaps in PMFL coverage, requiring manual fault isolation; (3) diagnostic testing is a time consuming process; (4) it is sufficiently complex to challenge maintenance personnel conceptually; (5) it is sufficiently large to demonstrate the principle (over 100 replaceable modules), and (6) it is sufficiently small to demonstrate utility in fault isolation.

EVOLUTION OF THE ARCHITECTURE

The developed architecture arose from the desire to increase the speed of accessing information within the expert system. Initially, a single knowledge data base (knowledge base) comprised of 3600 rules was used to describe the entire electrical relationship of modules with the CP Processor. With a SUN Microsystems, Inc. Model 3/110 computer, the access time for retrieving information in the full CP Processor model required 20 minutes in certain instances. Access time of this magnitude was deemed unacceptable for aiding technicians in the process of fault isolation. This led to the conclusion that a hierarchical architecture was necessary to reduce processing time, as well as to provide for eventual expansion of the trouble-shooting domain to the sonar system or combat system level.

As a test of the hypothesis that a hierarchical architecture would reduce processing times to an acceptable level, a simplified trial case was created. A subset of the knowledge base was created to describe a single channel, channel seven. The single channel knowledge base was compiled. Run time operations were then compared to similar operations conducted on the large knowledge base. Improvements were noted in the test recommendation and fault isolation times. On average, the test suggestion time range was $10 \text{ s} \leq t \leq 120 \text{ s}$. As a result, fault isolation was an order of magnitude faster, for the single channel trial case. Results from the single channel knowledge partition suggested that the entire CP Processor could be adequately described with a hierarchy which utilized channel symmetry and the presence of functional boundaries. Technician manipulation of obscure knowledge bases was perceived to be cumbersome and error laden. The solution chosen was to

provide a two layer hierarchy, with the initial layer encountered being the Local Area Expert (LAE), and the next layer being the Fault Isolation Layer (FIL). The LAE functions to direct maintenance to a single specific functional area, based upon information obtained by the technician from system indicators and PMFL. The FIL is subordinate to the LAE procedurally, but operationally is responsible for testing and maintenance procedures performed by the technician. Additionally, replacement modules are suggested by the FIL.

LAE

The LAE is itself an expert system whose primary function is to direct the technician to maintenance of a single functional area, and any additional functional areas until all malfunctions are removed. The LAE is not intended to isolate faults beyond the functional level.

The LAE performs the following tasks: initializing the knowledge state, updating knowledge, isolating to the functional area, invoking the FIL with the correct knowledge base, and passing the updated knowledge state to the FIL.

The LAE was constructed using C Code generated by the Rulemaster expert system shell as a basis. Then C Code was customized for efficiency and enhanced to facilitate integration with the FIL. Primarily, Rulemaster was used to incorporate performance monitoring logic presented by the AN/UYQ-21 display characteristics, Receiver Test Generator (RTG), PMFL and indicator lights. The Rulemaster generated code was cumbersome, in terms of efficiency and storage space, and circuitous in logic explanations. This software was streamlined to include the basic logic but reduce the layered explanations and procedural steps used to achieve functional isolation.

Procedural steps relate the various species of performance tests, e.g. the monitor's display of channel tracks is grouped as a single indicator test. Explanations of the logical path were condensed from the broad restatement of a diagnostic path to concise statements relating to functionality. The result was a concise expression of the knowledge presented by the sonar system's built-in fault isolation architecture.

Once the LAE is invoked by typing "lae" at the UNIX prompt, the user is queried for whether the knowledge state of a previous session should be loaded or a new session should be initiated. A continuation of a previous maintenance session recovers the knowledge state from a file written when the session was halted. Primarily, sessions which are interrupted by priority operational tasks, or which extend over work shifts are recovered.

New sessions commence with an examination of the PMFL indicators, as displayed on the Support Monitoring Unit (Unit 14). Only the PMFLs relevant to the CP Processor are displayed to the user. These are indicators 60-66, which correlate to

power supply modules, and indicators 294-300, which correlate faults with the various reference signals in the sonar system. The display of any of these PMFL indicators leads to the secondary layer of indicator light interrogations, which identifies functional abnormalities. The result of identifying functional abnormalities leads, through heuristic reasoning, to the selection of a single functional area to fault isolate.

The presence of a detected functional abnormality combined with absence of the associated PMFL fault code leads to another branch of logic. This branch considers the System Monitor as faulted and addresses functional isolation through other indicators, such as console display and the Receiver Test Generator (RTG). If appropriate it provides implications that supporting units outside of the CP Processor may also be faulted, and suggests that these supporting units be repaired before making further investigations into the CP Processor's functionality.

The result determined by the LAE states which function should be further fault isolated. Initially, if the LAE determines that multiple functional areas are faulted, the heuristics select a single function to interrogate. At the point of isolation to a functional entity, the user is presented with a textual description of the achieved result and an explanation of its consequence. The explanation is presented as "ADVICE" to the user.

Beyond the "ADVICE" presentation, the user is informed that the FIL is being invoked. Furthermore, the knowledge base, corresponding to the functional area believed to be faulted, is automatically loaded into computer memory.

From the first invocation of the FIL, the presence of the LAE is maintained only as a knowledge state tabulator. Primarily, the LAE function takes account of which tests have been performed by the FIL, the result obtained, any subsequent invalidations which occurred, and the present function being investigated. The FIL interchanges this information with the LAE at the conclusion of a fault isolation process. The LAE continues to redirect fault isolation to other functional areas, but only as a result of conclusions reached by the FIL. In general, the LAE would only actively participate in the selection of the first functional area isolation; beyond that, the FIL should have sufficient general information to redirect troubleshooting. The LAE only achieves the status of functional area selector if the FIL has insufficient knowledge to discriminate a new path (usually, an indication of a knowledge hole, or misconstruction of a functional area knowledge base).

Primarily, the LAE, under the normal conditions of well constructed knowledge bases, achieves the role of CP Processor functionality verifier. As functionality verifier, the LAE presents all branches of functionality logic and requests the user to verify that malfunction indications are not present. Barring the presence of a residual malfunction, the LAE terminates its maintenance involvement. Residual malfunction states would be resolved by the FIL, after the LAE heuristically selected the function abnormality.

FIL

The FIL is based upon the Fault Isolation System (FIS) Inference Engine, developed at the Naval Research Laboratory (NRL). The FIL possess enhancements of modified user interfaces, probability based heuristics (to recommend replacements), LAE integration software, and other subtle enhancements. The FIL's primary function is to isolate faults within a functional area to the replaceable module level. Component level faults are not considered. Isolation to the component level is not part of the Navy's maintenance strategy. The process performed by the FIL is summarized as follows: knowledge base is loaded into the FIL, user interacts with the FIL to achieve an efficient troubleshooting sequence, module replacement and redirection to other functional areas is recommended. The FIL is capable of addressing both single and multiple module faults.

Currently, the FIL is written in LUCID COMMON LISP for the SUN 3/110. There have been variants of FIS written for other computers and compilers, but the FIL variants do not exist. The FIL consists of two primary parts: the inference engine and the knowledge base. The inference engine is based on probabilistic reasoning. Implications result from each user update of information. The implications are summed to lead to a preponderance of evidence. Thus, entry of faulty information, while not ideal, does not lead to the FIL pursuing a path to incorrect module isolation. Instead, faulty information may increase the possibility that a module may be incorrectly identified as a possible cause, but correct information will adjust the probability accordingly.

In addition to calculating the probability of the various modules being faulty, the inference engine determines the most efficient path to obtain new information, based on the current state of disorder in the system knowledge and relative cost of diagnostic tests. The FIL primarily chooses each step in the path based upon what test yields the most knowledge on the state of the system (usually information on the greatest number of modules), or which test yield the most knowledge in the least costly manner. This primary selection factor is modified by the second factor, cost. Cost is provided as a priori knowledge and is based on the time required to perform a testing action. As a result, tests requiring extensive use of time are selected only if they will provide a relatively large increase in the knowledge state of the system.

All information used by the inference engine is either contained in the knowledge base or supplied by the technician, as a result of a FIL request. Primarily, the knowledge base contains a priori knowledge of the physical structure of the unit under test (UUT). The knowledge base consists of six unique groups of information. The rules are essential, since they express the causal nature of system diagnostics in relation to the physical topology of the UUT's circuitry. Causal rules were generated directly from the schematics of the CP processor. Test information contained by the knowledge base relates physical measurements to logical rule structures. Precondition information is used to activate and deactivate the appropriate rules when switches are thrown or the configuration is changed. Order information is essential

only if a group of tests must be performed in a certain sequence. Instruction information is presented in the knowledge base as a convenience for user interface, and is optional. If instructions exist, an index relating them to tests must also be present.

In the system architecture, the functional areas identified in the LAE must relate to a knowledge base for each functional area. Thus, since the LAE can identify twelve unique functional areas for the CP Processor, there are twelve different knowledge bases. The functional unit names are: left correlator, right correlator, center correlator, left channels, right channels, center channels, PMFL, power, timing, multiplexer reference, modulator reference and correlator reference. Overlap of the knowledge bases supports redirection of the testing effort. Each knowledge base represents adjacent functional areas as if they were replaceable modules; diagnosis of failure to a macromodule triggers redirection of the diagnostic effort to the appropriate knowledge base.

The structure of the FIL execution begins with the LAE invoking the FIL and passing the functional isolation information to the FIL. The information passed between layers is the chosen functional knowledge base to load and the available test result information. Upon loading the appropriate knowledge base, the test result information is used to update the system probability state. The initial system state consists of the module default failure probabilities, obtained from the normalized reliability data. The probability state is then adjusted by performing the information update, using the results passed by the LAE. All updating occurs before the interface is presented to the technician for continued fault isolation.

The interface presented, after information updating, is displayed in menu format, categorized into three areas: control, inspection and troubleshooting. Control provides commands to redirect the analysis to the LAE or other functional areas. Output device control functions also exist. The inspection menu provides tools for assessing the state of the system probability state. Two primary functions allow the user to make judgments on likely faults from their output, "show probabilities" and "ambiguity set". The "show probabilities" item provides a list of modules with their respective probabilities of being faulty and certified. Probabilities of being faulty are accumulated from the result of a failed test and weighted by the logical proximity of the module to the testpoint. Modules in the ambiguity set of a faulted test have their fault probability raised. Other modules may not have their fault probability impacted or may have it lowered. Certification is the result of a weighted probability based upon the total number of module tests and the logical proximity of the test to the module. Certification is accumulated by all modules in the ambiguity set of the test; other modules do not gain certification from the test result. A condensed presentation of the ten most probable module failures is also available in bar-graph form.

The ambiguity set command is available to inspect the modules in the ambiguity set of the last faulted test. Together with the probability information, the user gains the capability to make judgment replacements at any point, based upon the

information presented and experience. Other functions are available in the inspection menu to display the list of tests which have been performed.

The troubleshooting menu is used most often, since it directs and selects testing through "best-test" and "make-test" commands. The ability to remove the result of a test is also available. The "best-test" command promotes efficient testing, based on the heuristic drive to increase system "knowledge" and cost. The user has the option of accepting the recommendation, or specifying a different test using the "make-test" command. For reference, instructions are presented to the user. The user performs the physical test in either case and enters the result. After each testing action the probabilities, certifications, and system probability state are recalculated.

Testing continues until some form of resolution is reached. One possibility is for the user to choose to make a replacement based on the information available and experience. Another case is that the system isolates a fault to a module or modules, and the technician performs the replacement. The final possibility is that the FIL isolates the fault to a macromodule, a condensed logical representation of the knowledge base for another functional area. In the first two cases, the user has the option to continue troubleshooting (usually to confirm functionality) or to return to the LAE. In the last possibility, troubleshooting is redirected to the other function knowledge base by loading the new knowledge base. Troubleshooting continues. All of the cases involve the interaction of the LAE and FIL at an interface.

INTERACTION OF LAE AND FIL

The interaction of the two layers can simply be described as follows: the LAE is used to perform isolation to the functional area, and the FIL is used to perform fault isolation to the replaceable module. Figure 1 indicates the flow of control. Referring to figure 1 and the previous LAE section of this paper, the flow of the fault isolation process begins with the sonar system technician detecting an error state in the CP processor. The LAE is invoked and makes inquiries into the state of system fault codes. Isolation is made to the functional level. The LAE then invokes the FIL and passes two information items: the name of the functional knowledge base which should be loaded and a list of test information obtained.

The FIL loads the suggested knowledge base and initializes the system state with the known test results. The FIL then continues the testing at the module level by assisting the technician in choosing the most efficient tests. Testing continues until the technician opts to replace a module using their experience and the information obtained from the FIL, or the FIL, automatically, suggests a replacement.

When a decision is made to perform a replacement, those test results related to the module replaced (those who have the module in their ambiguity set), are invalidated. Tests whose validity does not depend on the module replacement remain

valid. The FIL passes all the information on valid and invalid tests to the next cycle of fault isolation.

The next cycle of isolation has two different paths depending on the type of replacement. If a single or multiple module replacement is made control is returned to the LAE to affirm the functionality of the CP Processor. If the LAE confirms the functionality of the CP Processor, the system exits to UNIX. If additional functional errors are found by the LAE, the cycle begins again. The second path occurs if a macromodule is suggested for replacement. In this case, the LAE updates its database of exercised test results and reinvokes the FIL with the appropriate knowledge base. The cyclic reasoning continues until all malfunctions in the CP Processor are eliminated.

VALIDATION

Validation of the TAS occurred over several stages in the development. Initially the knowledge database was correlated with the schematic diagrams to account for inaccuracies in the causal logic. Upon compilation of the knowledge base into the FIL form, diagnostics were presented for the detection of syntax and gross format errors. After compilation, utilities to examine rule and ambiguity consistencies were used to examine the knowledge linkage. Finally, prototype simulations were executed to test the convergence of the knowledge bases.

Upon having obtained a great deal of confidence in the knowledge integrity, prototype field testing was performed at the Naval Underwater Systems Center, AN/SQS-53B test bed in New London, CT. One particular fault was diagnosed during the demonstration of the prototype TAS - a fault in a delay line of the CP Processor. While full testing and validation of the TAS still remains before final implementation, we felt that the prototype demonstrated a high degree of feasibility for the application to a total troubleshooting architecture.

CONCLUSIONS

A two layer architecture was demonstrated to address the fault isolation requirements for the CP Processor of the AN/SQS-53B. The architecture incorporated one layer to perform functional isolation and another layer to perform isolation to faulty replaceable modules. The prototype exhibited both logical consistency and the capability to isolate physical faults during the validation and testing phases. The architecture is generic, which allows it to be extended to the entire sonar system or even beyond, to several ship systems. The main requirement for specific heuristic information, at the functional level of isolation (LAE), is viewed as the only tailoring required to extend the complexity coverage of the system.

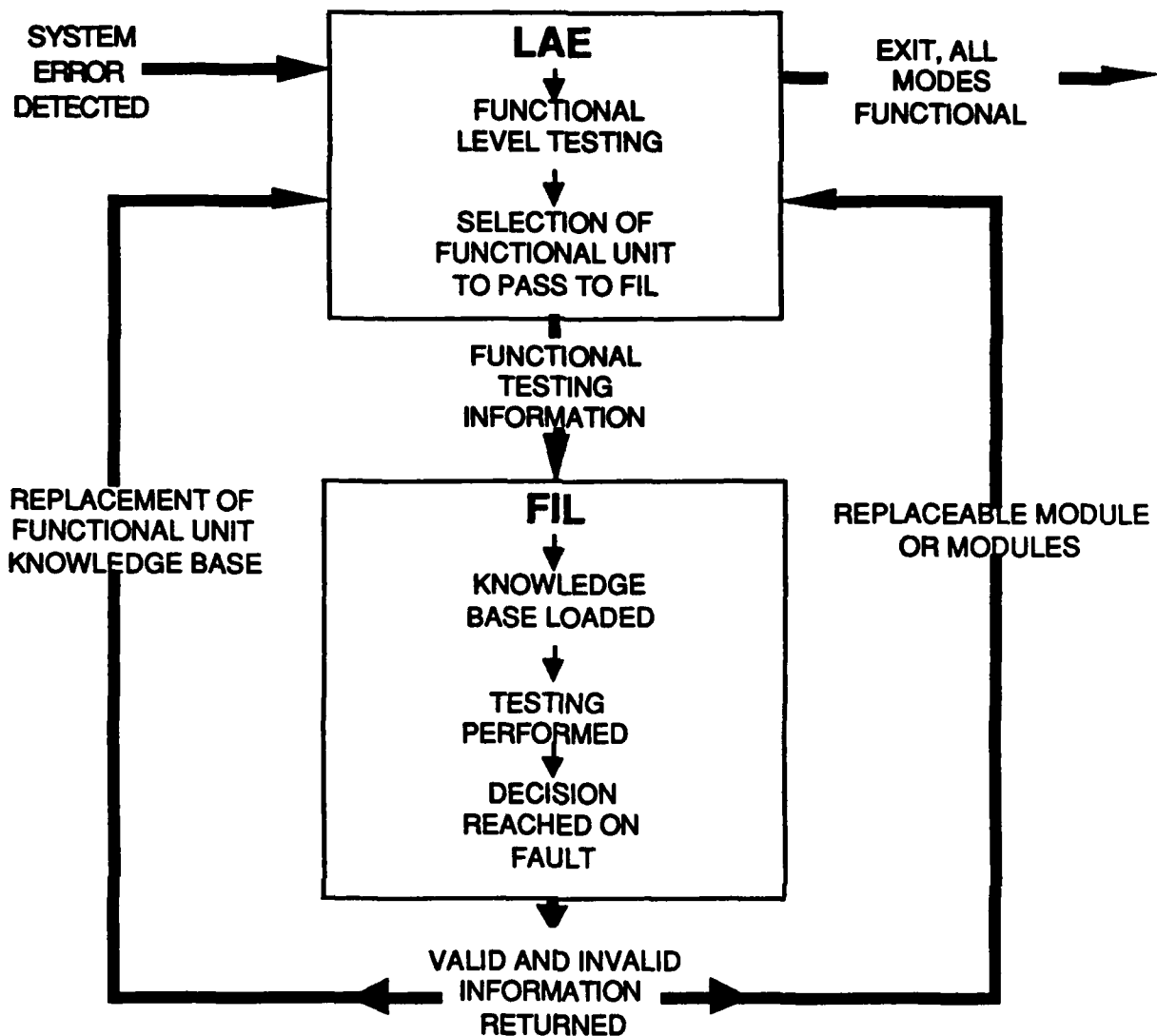


Figure 1. Representation of the TAS hierarchy, displaying the major system functions of the LAE and the FIL and the interface process.

ACKNOWLEDGEMENTS

We would like to thank the Naval Sea Systems Command, PMS411, for their support and interest in addressing the problem of electronic system maintenance through the use of Artificial Intelligence techniques; Dr. Frank Pipitone and Mr. William Spears for their advice; and Dr. John Davis, Dr. Randall Shumaker, Mr. Lee Kline and Mr. Thomas Gattis for their enthusiasm for the project.