

**STATUS OF THE ADVANCED AVIONICS SUBSYSTEM TECHNOLOGY (AAST)
FAULT TOLERANCE DEMONSTRATION**

Timothy P. Monaghan

Naval Air Warfare Center Aircraft Division Warminster, PA 18974-0591

ABSTRACT

The Advanced Avionics Subsystem Technology (AAST) Fault Tolerant Demonstration will clarify the Navy's fault tolerant avionics specifications, validation methods and acceptance tests. This paper will describe the status of this program. The program is in the process of demonstrating clear and precise fault tolerant requirements for specifications, the Statement Of Work (SOW), and the series of validation metrics and acceptance tests, from initial concept exploration to the brassboard Dem/Val system, as the government oversees the design and procurement of a modern complex computer based weapon system. This language and process are documented in the first draft of "Specification and Validation of Fault Tolerance in Electronic Systems Development".

TRENDS AND DRIVERS

The DoD is entering a radically new era of systems procurement. The Cold War is over and new scenarios and radical force structure changes are changing the requirements of the next generation of weapon systems. Other factors in the "new world order" of weapon system procurement are: budgets are decreasing, development cycles are being stretched out, prototypes and demonstrations are being emphasized, open systems standards will dominate, and the next generation of systems will be intensively modeled and simulated before any hardware is built. Future complex weapon systems will increasingly rely on digital systems and the dependability of these digital components will play a critical role in the effectiveness of those systems in the field. The frequency of hard fault rates is decreasing for both military and commercial digital parts but transient fault rates are increasing. The next generation of systems need to automatically handle transient faults and notify the operator only when recovering from hard faults yet record all fault activity to ensure quick diagnosis of failing parts. During the Gulf War, the fleet aircraft

sustained high Full Mission Capable (FMC) rates (in the high 90% range), but these high FMC rates were sustained at the cost of high Maintenance Man Hours per Flight Hour (MMH/FH), (30 to 65 MMH/FH), high spares usage and high false alarm rates, (consistently in the 30% to 35% range). In a longer term combat situation, these factors; the rate of spares usage, the time to isolate and repair the equipment, and the false alarm rates will play a critical role in the effectiveness of the next generation of complex, computer based weapon systems.

The key issue that the AAST Fault Tolerant Demonstration is addressing in this new era of defense procurement is - how can the Navy manage and procure dependable and cost effective, computer-based weapon systems? The demonstration program will investigate the timely and practical application of fault tolerant technology early in the design cycle before major resources are committed to a particular design. This application of fault tolerant technology will be balanced against the extreme time pressures of modern avionics system development.

THE ADVANCED AVIONICS SUBSYSTEM TECHNOLOGY FAULT TOLERANT DEMONSTRATION

Funds for the research, development, transition and insertion of new technologies into the fleet are divided into 6.1, 6.2, 6.3A and 6.4 funds. The 6.1 and 6.2 funds are focused on exploring the feasibility of new technologies. The 6.3A funds are aimed at demonstrating those technologies so that program offices can specify them with confidence. In 1990, ONR's 6.1 research started the Ultradependable Multicomputers and Electronic Systems Research Initiative. This research initiative addresses a wide ranging number of fault tolerance topics including measurement and modeling of expected system fault modes, fault injection, simulation and modeling techniques, software fault tolerance approaches, as well as compiler, algorithm and hardware-based fault-tolerance techniques. ONR's 6.2 exploratory development, computer block has an effort called the "Engineering of

Complex Systems Technology" whose aim is to explore the entire design and development of advanced real-time systems. The fault tolerance portion of the block plan is aimed at integrating fault tolerance into the design process of complex systems.

The Advanced Avionics Subsystem Technology (AAST) Fault Tolerance Demonstration is a 6.3A project that takes the 6.2 Engineering of Complex Systems effort the next step and demonstrates the fault tolerance metrics and acceptance tests at each stage of an evolving contractor's design. The AAST work will also transition some of the ONR 6.1 developed tools (fault injection, fault tolerance benchmarks and fault tolerance simulation techniques). The goal of the AAST Fault Tolerance Demonstration is to demonstrate the necessary and sufficient dependability metrics and validation techniques of a fault tolerant system. These requirements will be documented so that program offices can use them in their specifications and SOW packages according to their various fault tolerance and dependability needs.

LANGUAGE AND PROCESS

The two key thrusts of the AAST Fault Tolerance Demonstration program are: what legal language needs to be in the SOW and Specification to clarify to the contractor the exact fault tolerant and dependability requirements that the government expects to see in the system? And what is a model of the process for the government and contractor to engage in to ensure that the system will be dependable?

The dependability contract language in current requirements documents tend to ask for very general and vague qualities like - "the system shall be fault tolerant". For instance, a current SOW requires: "The contractor shall identify and discuss high risk areas..." and "single point failure modes that significantly degrade mission performance or prevent successful mission accomplishment shall be addressed...". This particular SOW also provides the traditional reliability, availability, and maintainability numbers and reconfiguration, degraded modes and testability requirements. But very few of these requirements translate into a directly testable quantity on the design like a benchmark. In the extreme time pressures of modern complex weapon system design vague requirements like fault tolerance are too easily put on the back burner and simply not given enough attention. The AAST Fault Tolerance Demonstration program seeks to make the fault tolerance and dependability features of a system more visible and quantifiable.

Legal, formal, specifications are the only way the government can define computer performance and dependability requirements. More precise fault tolerant specifications would specify the system's fault containment regions, the specific faults the system will guard against, how these faults will be characterized (timing, extent and duration), and the types of analysis and fault injection testing expected to be done at each stage of the system design. In the automatic reconfiguration requirement, the specifications should state that the system shall automatically reconfigure to circumvent a failed component and select a back up component or data path to redistribute the processing. This reconfiguration shall be transparent to the operator and the operator should have the ability to be notified of reconfiguration from hard faults. The system should also be capable of automatic and transparent reconfiguration of the system when a transient faulted component is restored to operation. This process should continue as long as the processing and component resources continue to meet the demands of the reconfigured system. When system resources are inadequate to meet the real time system demands, the system should enter the degraded mode state. The degraded mode specification should basically state that the system should never just "give up" when system resources are not at an optimal level. The system should still perform if able to do useful things. The specification should require that the multiple processors and the distributed processing architecture shall permit the automatic, controlled and graceful degradation of system capabilities when errors occur and the real time processing demands exceed system resources. Functional capabilities shall be automatically redistributed to the rest of the system according to mission priorities. Impact on mission operations shall be kept to a minimum for as long as possible when operating in the degraded mode.

The SOW is the requirements for the contractor design team to fulfill at each stage of the system design. Generally, the SOW should require that the error handling features of the system shall be validated at each stage of the systems evolution. The validation should be a functional fault analysis which will map the specified fault set onto each subsystem partition and then identify the fault detection, isolation, removal and recovery mechanisms of the system that will enable the system to maintain the mission services in the presence of faults.

Each design review requires specific exit criteria for that design stage. The

dependability requirements for each design review should be clarified. For the System Design Review (SDR), for example, the exit criteria might be:

Exit Criteria for SDR

- Preliminary architectural description, architectural partitioning and functional allocation
- A system development plan including:
 1. Modeling tools for the evaluation of system operational performance and system fault tolerance
 2. Identification of a baseline set of application based benchmarks
 3. Identification of critical state information which must be protected under all fault conditions for transparent fault recovery

Besides precise legal language dealing with dependability and fault tolerance, DoD needs a clear model of the process of contractor analysis and design choices and government evaluations. DoD needs to be able to specify clear and quantifiable validation techniques at each stage of the system's design that allow DoD to be informed customers able to quantify a design's dependability and fault tolerant metrics and reasonably ensure at each stage of the design that the evolving system will be a dependable system to own and operate.

THE PROCESS - AN OVERALL MODEL FOR MONITORING THE DESIGN OF A DEPENDABLE SYSTEM

The fault tolerant validation of a computer system involves both analytic modeling and fault injection.

The basic elements of analytic modeling use the input parameters such as the failure rates, coverage estimate, repair rates, and costs of repair and parts. The analytic system behavior modeling elements can be combinatorial or non-combinatorial. The combinatorial elements are reliability graphs, predicting the hard failure rate of the current system from the statistical past failure rates of similar components and fault trees, trees of conditions that lead to certain system failures. The non-combinatorial modeling elements are Markov chains, a graph of failure states and probability of transitions that determine the probability of the next state of the system. (Markov models are good for sequence dependent failures and repairable systems.) Petri nets are useful when the limits of Markov chains are reached. Petri nets can be used to examine inherent concurrencies such as coincident faults (a second fault arriving while the system is recovering from the first fault) and Monte Carlo simulations are useful as more

system reality is put into the model and the analytic solutions become intractable. [1]

The component failure rate data and fault models used to model a new system are the component failure rates and failure modes of past similar systems and similar components. From this measured data of past similar systems, the modeler predicts the future fault rates and models the assumed fault handling of this system. The initial Markov, Petri or combinatorial models of the system are thus the initial and gross assumptions on how this new system will probably handle faults in the future. However they are useful in the initial stages of design when the designer needs an estimate of reliability.

Fault injection experiments are performed on the functional simulation of the system that develops from the high level conceptual model of the system and is gradually transformed from behavioral descriptions to RTL descriptions to logic and circuit descriptions to the final brassboard implementation.

As the design is further refined, faults are injected into the current functional representation of the system. The results of the fault injection experiments are used to validate and refine the analytic model. Figure 1 shows this overall flow and feedback from field data to analytic model to fault injection experiments on the current functional model (VHDL behavioral and structural). The analytic model is useful as the functional design evolves. The functional model can never be exhaustively tested but a further refined analytic model reveals which system parameters are most sensitive to change and need closer functional fault injection experiments. Thus, the analytic modeling can enable the test engineer to wisely inject faults in the short amount of time allowed for the fault injection testing.

Random fault injections should also be carried out to allow for any fault handling assumptions that have been overlooked in the analytic model. The overriding goal should be an adequate test set at each stage of a system's design, given the amount of time allocated for testing, to ensure that the evolving system will be dependable and continue to deliver its services in the presence of transient and hard faults.

For another view of the same process, consider analytic modeling and fault injection experiments as two complimentary streams of dependable system development. (Figure 2.) Analytic modeling provides the insight to the fault injection experiments on which fault/level/system states to inject the faults

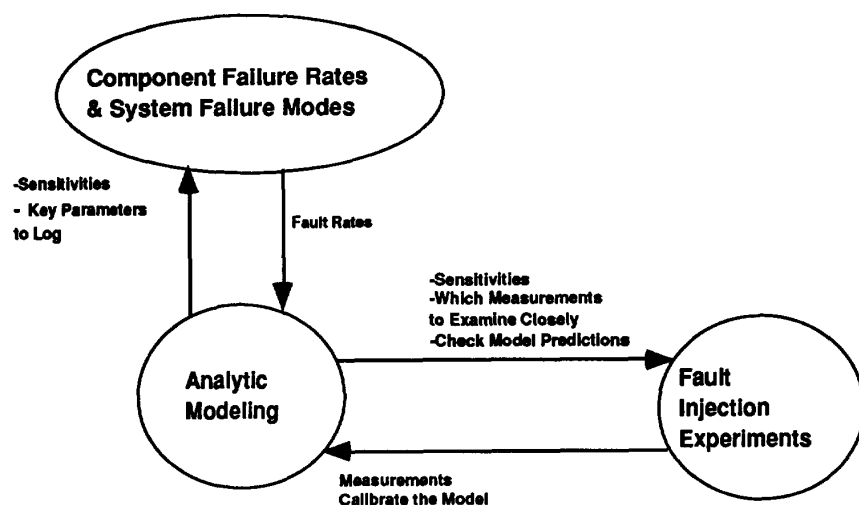


Figure 1

and the fault injection experiments calibrate the parameters of the analytic modeling. Thus, the results of both streams are constantly being reconciled and the estimates of dependability are becoming more accurate as the

design approaches a brassboard implementation. Carefully planned fault injection experiments in the brassboard are the final proof of the systems dependability.

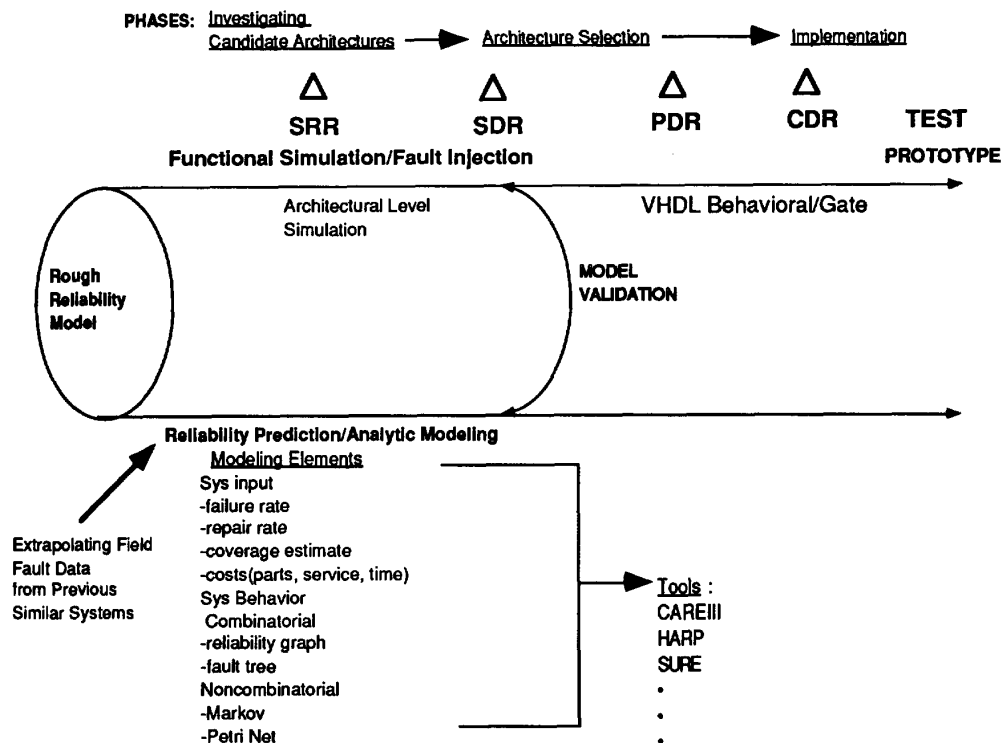


Figure 2

FT HANDBOOK "SPECIFICATION AND VALIDATION OF FAULT TOLERANCE IN ELECTRONIC SYSTEMS DEVELOPMENT"

The AAST Fault Tolerant Demonstration currently has a draft document that codifies the results of our initial investigation into the adequate Specification and SOW language for a dependable system and the model of the government and contractor interaction and the overall design process that will ensure a dependable system is designed. The document will include for example, Spec/SOW Language for different levels of system fault tolerance. Thus, depending on your overall system fault tolerant and dependability needs, a program office can pull adequate language out of the handbook and adjust it accordingly. The handbook also includes guidance to the program office on the kinds of analytic modeling and fault injection experiments needed at each stage and the government and contractor interaction for each milestone. The document is currently being examined by a wide number of industry, university and Federally Funded Research Centers (FFRC) reviewers. For a copy of the current draft, contact Liz Piergiiovanni, (215) 441-3281, email: piergio@nadc.navy.mil.

THE 2-C MISSION COMPUTER UPGRADE

The E2-C, the fleet's surveillance aircraft, is procuring an upgrade to the mission computer. This system will be an entirely new mission computer able to handle the aircraft's intensive computing needs for the remainder of the aircraft's life cycle. This program presents an opportunity to focus the AAST fault tolerant demonstration in a way which immediately benefits a platform while defining generally applicable fault tolerant metrics for other programs. The AAST Fault Tolerant Demonstration program has provided the program office with the precise fault tolerant and dependability language for the E2-C Specification and SOW package. This language has come from the fault tolerant community through the tri-service Dependability Working Group (DWG). The DWG is composed of leading researchers and industry developers of fault tolerant computer systems who, under the leadership of the DOD, are addressing the topic of the dependability validation of military computer based, weapon systems. The goal of the task group is to coordinate industry and the research community to come to a consensus on: 1) the specifications of the key dependability factors in computer based weapon systems and 2) the necessary and sufficient dependability validation criteria for these systems. The AAST Fault Tolerant Demonstration will assist the E2-C program office's procurement of the mission computer upgrade to ensure that the system is fault tolerant and

dependable yet the fault tolerant requirements are not unrealistic given the extreme time and cost constraints of modern avionics system procurement.

TOOLS AND TECHNIQUES SUPPORTING THE AAST DEMONSTRATION

The Fault Tolerance Handbook "Specification and Validation of Fault Tolerance in Electronic Systems Development" will not specify any specific tool set but in order to investigate the feasibility of the analytic modeling and fault injection modeling of the Specification and SOW requirements that are the output of the program, the AAST Fault Tolerance Program is working with the following tools and system validation techniques.

Professor Flaviu Cristian of the University of California San Diego is under contract to develop a methodology for specifying fault-tolerant systems so that those systems will support their systematic validation through fault injection. This work will guide the government validation of the system and enable the government to construct fault injection experiments that will reveal any defects in the dependability and fault tolerant aspects of the design. The goal of this testing will be to exercise the recovery/exception handling algorithms as early as possible in the system design before major resources are committed to a particular design implementation.

Dr. George Gilley of the Aerospace Corporation will assist the AAST Demonstration to implement his "Functional Fault Analysis" [2]. This analysis applies the specified fault set to each of the systems fault containment regions, determines if that region will detect that fault, what error reporting signal will be generated, the impact and latency of the fault, and the system's fault isolation and recovery functions for that error. Functional Fault Analysis is basically a check list of questions for the government technical team to ask the system design team as they walk through each segment of the design.

The AAST Demonstration will be using three fault injection tools: FOCUS and SYDA from the University of Illinois and FERRARI from the University of Texas. FOCUS is a simulation environment for fault sensitivity analysis of IC chips [3]. FOCUS adds a current source to target node and alters the voltage level over the time interval of the experiment. SYDA models a system at the processor-memory-switch level and also models software execution on the processor-memory-switch model. A fault dictionary will allow manifested faults to be simulated and evaluated at the system level.

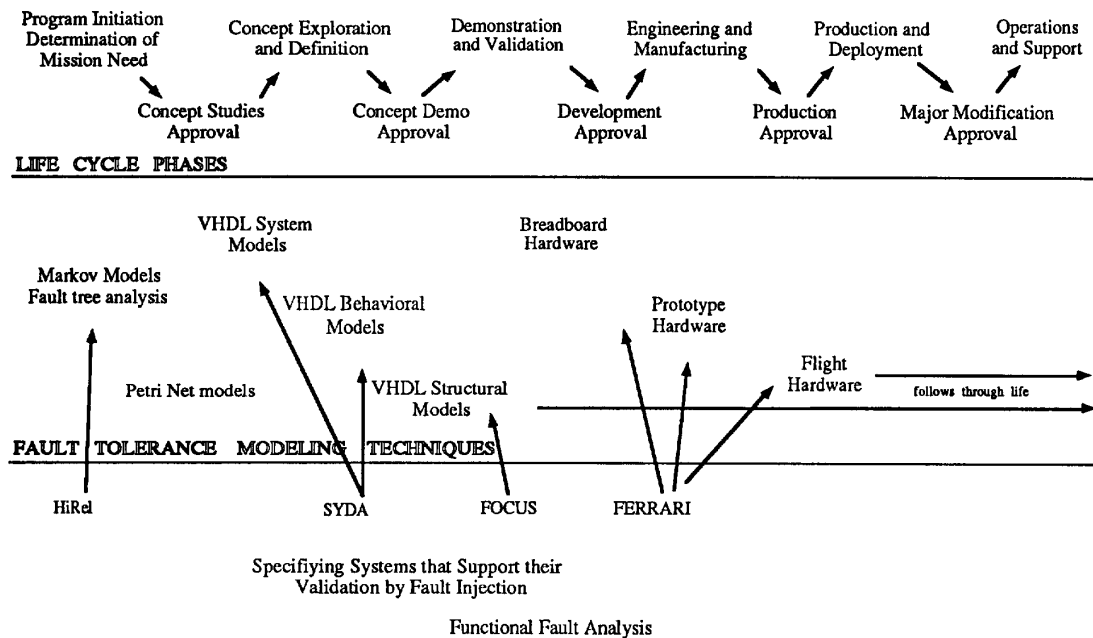
FERRARI is a software fault injector that operates on a breadboard or the final hardware implementation of the system [4]. FERRARI can inject hard or transient faults at the register level and can inject those faults while an actual real-time task is running on the system under test.

We are also using the Hybrid Automated Reliability Predictor Integrated Reliability Tool System (HiRel) developed by NASA Langley Research Center. HiRel is an analytic modeling tool that allows two types of behavioral models of a system: (1) the fault occurrence/repair

behavior of a system, containing information about the hardware redundancy, the fault arrival process and the manual (off-line) repair of the system and (2) the fault error-handling model which models the on-line recovery procedures for permanent, transient or intermittent faults. HiRel accepts inputs as either Markov models or fault trees. If a fault tree is the input it is converted to a Markov chain [5].

Figure 4 maps HiRel, SYDA, FOCUS and FERRARI, onto the design reviews and life cycle phases of the AAST Fault Tolerance Demonstration.

TOOLS & TECHNIQUES Supporting the AAST Demo



FAULT TOLERANCE ANALYSIS TOOLS

Figure 3

REFERENCES

- [1] A. Johnson, M. Malek, "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability", ACM Computing Surveys, Vol. 20, No.4, Dec. 1988.
- [2] G. Gilley, "Validation of Fault Tolerant Designs," IEEE/AIAA 10th Digital Avionics Systems Conference, October 1991.
- [3] G.S. Choi, R.K. Iyer, "FOCUS: An Experimental Environment for Validation of Fault Sensitivity Analysis," IEEE Trans. Computers,

Vol. 41, No. 12, pp. 1515-1526, Dec. 1992.

- [4] G.A. Kanawati, N.A. Kanawati, and J.A. Abraham, "FERRARI: A Tool for The Validation of System Dependability Properties", 22nd International Symposium on Fault-Tolerant Computing, 1992, pp. 336-344.

- [5] S.J. Bavuso, J.B. Dugan, K. Trivedi, B. Rothman, M. Boyd, "Applications of the Hybrid Automated Reliability Predictor", NASA Technical Paper 2760, Dec. 1988.