Chapter 5 – MISSION MANAGEMENT AND ROBUSTNESS

5.1 INTRODUCTION

Autonomy is often praised as a technology capable of reducing cost and enhancing performance in mission operation and management. Through careful deployment within the overall mission architecture, automation can augment or replace human decision making in order to increase reaction speeds, reduce errors and stress, mitigate cognitive overload, enhance safety, lower costs, focus analysis, cut bandwidth requirements, and free human reasoning for strategic tasks requiring high levels of robustness. Here, system health management is a specific mission operation task that provides the robustness. Through the use of automation and fuzzy algorithms the mission tasks can be enhanced further.

5.1.1 Risk Management

The role of risk assessment and risk management is to continuously identify, analyze, plan, track, control, and communicate the risks associated with an integrated system. Risk in itself, the possibility of suffering a loss, should not be avoided, and rather is essential to progress because failure is often a vital part of learning. Managing risk is the key to success, so that it is beneficial and not detrimental. In the context of software engineering and development, risk can be defined as the possibility of suffering a diminished level of success (loss) within a software–dependent development program. This prospect of loss is such that the application of the selected theories, principles, or techniques may fail to yield the right software products. In terms of the hardware, the risk in failure, unavailability of the hardware due to schedule or unavailability of expert manpower also contributes to risk. Resources available manpower, and schedule delay all contribute to risk. Table 5.1 in basic terms indicates the differences and consequences among risk assessment, risk abatement and risk management. Risk Assessment is an essential element of risk management. Once a risk has been recognized and evaluated as being important, steps must be taken to minimize the risk. Risk abatement is the main strategy for managing risk.

Table 5.1: Risk Assessment, Abatement, and Management

Risk Assessment	Risk Abatement	Risk Management
What can go wrong?	How we can prevent from happening?	What can be done?
What is the likelihood that something could go wrong?	What needs to be done?	What are the available options and tradeoffs?
What are the associated consequences?	Cost and schedule changes?	What are the impacts of current decisions to future options?
Adverse effects?	What option is the best option?	What is the best option? At what price?

Figure 5.1 describes typical top-level actions that are common to all methods in conducting risk assessment. This process includes risk category or areas such as cost, schedule, and performance. The process includes a core set of assessment tasks and is related to the other two categories. This requires supportive analysis among areas to ensure the integration of the assessment process.

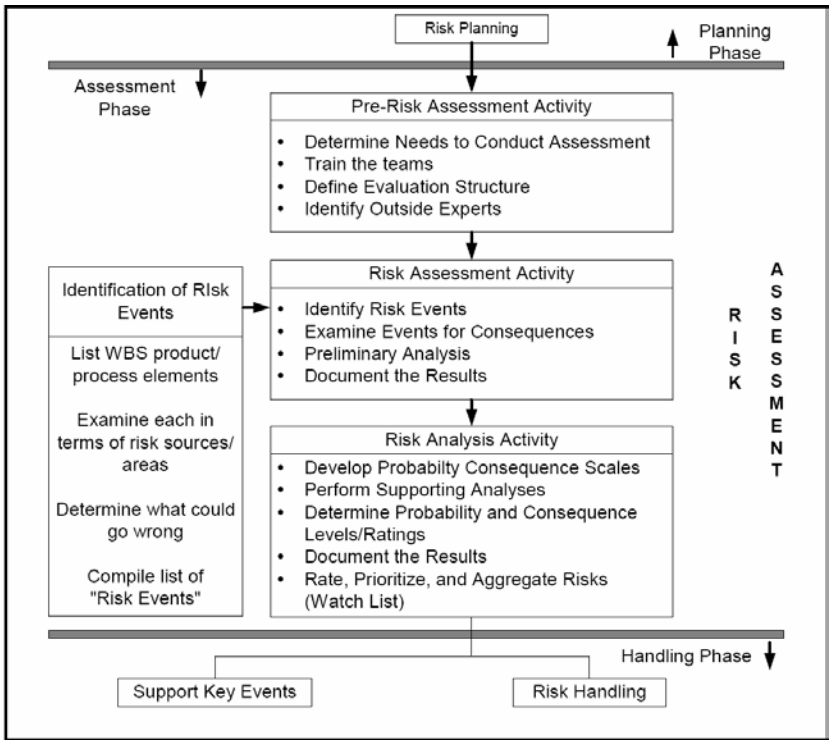


Figure 5.1: General DoD Risk Assessment Process [Anon, Risk Management Guide for DoD Acquisition].

5.1.2 Uncertainty Management

Uncertainty management is a significant long-term foundational issue for planning, design and management of engineering systems. Given the spatial and temporal uncertainty associated with battlespace visualization, uncertainty management must be an integral component of future systems designed to aid commanders and staffs in developing, disseminating, and executing a commander's concept of the operation. Problems ranging from control of the UAVs, to spacecraft formation right for planetary exploration, to supply chain management are all examples of systems in which networked controls, communications, and computation are playing an increasingly integral role. All of these problems: distributed computation, cooperative planning, and uncertainty management, present challenges for which the existing theory leaves many important questions unaddressed. Current tools for analysis and design of flight control systems are focused on control of individual vehicles or small numbers of coordinated vehicles. While these tools allow sensing and communication channels to be included, there is not yet a deep understanding of the role that information now plays within a networked control system, especially as the number of vehicles increases. There has been some recent progress in this area for networked control systems; but, there has not yet been an exploration of the fundamental limits imposed by sensor and communication topologies (e.g., what is the minimal information needed between sets of vehicles in order to maintain stable formations in the presence of uncertainty). Early work on string stability in longitudinal control of automobile platoons showed that this information now was important and provided insights for the one dimensional case. This situation becomes much more complicated for aerial vehicles operating in dynamic, uncertain and adversarial environments. In the absence of a centralized controller, the Formation Self-Assessment (meaning determining its own properties) is necessary. Examples of self-assessment include health monitoring and role management. When the information flow topologies are themselves uncertain and dynamic, a crucial component of formation self-assessment is the determination of those topologies to enable implementing other distributed algorithms which act along those communication networks. What the above challenges have in common is

that they involve managing the interaction of an intelligent, dynamic, and networked system with a dynamic, uncertain, and adversarial environment in a setting where the means to manage that interaction are constrained by limitations in the flow of information within the network. Accordingly, we have identified the following areas to pursue to in our research. Taken together, these different research goals address aspects of cooperative control of multi-vehicle systems which we consider to be crucial in enabling this technology. Our overarching goal is the implementation of a control architecture and control laws which are capable of achieving mission objectives and reacting intelligently and robustly to the uncertainty in the environment while maintaining the formation's ability to communicate with itself. We intend to proceed incrementally toward this admittedly ambitious goal, emphasizing the need to clearly articulate a framework through which these questions can be analyzed, and leveraging our research experience in dynamical systems theory and control of individual vehicles wherever possible. In the context of autonomous flight control, the adaptive model-based propulsion control system can deliver the desired thrust response to the vehicle management system where a pilot might otherwise have needed to manually adjust the throttle. This is particularly important for an aircraft with multiple engines, since asymmetric thrust response can result in an unacceptably large yawing moment. Additionally, the sluggish thrust response of a degraded engine would be an uncertainty for an autonomous vehicle management system performing transient maneuvers.

5.1.2.1 Definitions of Uncertainty

Uncertainty is a general concept that reflects our lack of sureness about something or someone, ranging from just short of complete sureness to an almost complete lack of conviction about an outcome. Uncertainties exist on sensor fault, output signal, and models. This uncertainty can be viewed as output multiplicative uncertainty. DeLaurentis and Mavris provide the most fitting definition for our application: "Uncertainty is the incompleteness in knowledge, either in information or context, which causes model-based predictions to differ from reality in a manner described by some probability distribution function." Uncertainty categorized by various sources and can be described in Table 5.2. Each of these uncertainty types must be considered to maximize the probability of success or robustness. When uncertainties in consequence (performance, cost, and schedule) are calculated in a design uncertainty analysis, the resulting consequence probability distributions are a complete measure of risk.

Table 5.2: Types of Uncertainty [Alan Brown and Timothy Mierzwicki]

Types of Uncertainty	Description
Inherent Uncertainty	Uncertainty due to the variability inherent in a system design, technology or the environment
Statistical Uncertainty	Uncertainty due to the incompleteness of statistical data
Modeling Uncertainty	Uncertainty resulting from the simplification of nature and physics
Human Uncertainty & Error	Uncertainty due to differences of opinion (or subjective uncertainty) and misdiagnosis (or diagnostic uncertainty) including: errors in calculation; selection of the wrong known data; inadequate design review; failure in calculating critical conditions; poor quality fabrication; use of the wrong materials; and abuse by operators.

There is a significant need for data integration capabilities in land, air and sea, which has manifested itself as products in the multination defence infrastructure. However, in dealing with flight and engine or perhaps system data it has become apparent that existing data integration products do not handle uncertainties in the data very well. This leads to systems that often produce an explosion of less relevant answers which subsequently leads to a loss of more relevant answers by overloading the operator of that

machine. How to incorporate functionality into data integration systems to properly handle uncertainties and make results more useful has become an important issue. Data uncertainty results from two different sources:

1) Inherent data uncertainties are attributes of the data itself and not artifacts of its representation. Data generated from laboratory experimental methods often have inherent uncertainties.
2) Data Representation Uncertainties. Data representation uncertainties result from the mapping of real world information onto a computable representation of this information.

In general, uncertainty describes the extent to which a system environment is known and understood. Efficiency points to how well data can be communicated with the systems. Another important issue that must be accounted for in the design of model-based fault diagnosis and fault-tolerant control systems is the presence of model uncertainty (also referred to as the mismatch between the system and the model). Model uncertainty was a key motivation for introducing feedback and that classical control theory had very effective ways of dealing with uncertainty both qualitatively and quantitatively. Process uncertainty could be described very easily as a variation in the process transfer function with the caveat that the disturbances do not change the number of right half plane poles of the system. A good starting point for discussing deviations between model estimates and observations is a conceptual framework.

Let us consider the situation when the parameter set α has a certain set of values [H.R. Oleson]. Let us assume that our model is deterministic and thus purports to predict one number for each α: the ensemble average (α) °C for a large number of realisations. If we consider just one realisation, then the observed concentration (Co) can be decomposed as follows:

$$\underbrace{C_o(\alpha,\beta)}_{\text{Observed concentration}} = \underbrace{\overline{C}_o(\alpha)}_{\text{Ensemble average}} + \underbrace{c(\Delta c)}_{\text{Measurement error}} + \underbrace{c(\alpha,\beta)}_{\text{Inherent uncertainty}} \qquad (1)$$

Note that the decomposition depends on the definition of α and thus is model-dependent. This is an annoying fact, which we must accept. The “measurement error” term accounts not only for trivial instrument inaccuracy, but also for the fact that we may not measure the parameter that we assume. The measurement error can in principle be reduced by increasing the number and the accuracy of measurements. A way to avoid unnecessarily severe effects of measurement error is through the use of quality indicators assigned to data, so that misleading observations can be discarded. The key parameter is how we can make a decision based on the data and knowledge gained.

Figure 5.2: From Data to Knowledge.

The disclosed systems and methods are directed to exchange of uncertainty information between interacting modules. Variances in the outputs of one module may be required as uncertainties in the inputs of another module. The disclosed systems and methods allow for simple and efficient communication of the uncertainty information between any modules. In one aspect, a system of integrated modules includes one or more application modules and at least one interface module. The application modules are adapted to receive inputs and/or generate outputs, including uncertainty distribution information. The interface modules are adapted to communicate with one or more application modules and to translate uncertainty information between a predetermined uniform format and an application-specific format.

The key idea is to decompose, respectively, observed concentrations and modeled concentrations. It is a basic assumption that we have a model formulated in terms of a set of input. A good starting point for discussing deviations between model estimates and observations is a conceptual framework, which has been described in several papers.

5.1.2.2 Sources of Uncertainty

Uncertainty sources might be characterized in terms of system level axis. At one end might be uncertainties due to the dynamic model stability and control derivatives. At the other end might be mission type environmental factors like numbers of targets or false targets. Somewhere between these extremes might be damage recovery effectiveness and losses due to enemy defences. Also, it might be notable that uncertainty reduction can be a cause for consideration of coordinated, multi-asset solutions to begin with. Rationales for multi-asset solutions are that:

1) There are things that can be done with multiple assets that simply cannot be done with one; and
2) There is greater flexibility and inherent fault tolerance with multiple assets.

For example a problem requiring saturation of an enemy defence cannot be done with one asset.

5.2 REVIEW OF CURRENT APPROACHES

The engineering of mission management knowledge based processes for the command of multiple intelligent autonomous vehicles for land, air and sea is the coordination of elements of a distributed system so as to generate coherent behaviour and robustness. As such the techniques apply to the management and control of military command and control. The ever increasing demand for cost effectiveness, project efficiency and increasing productivity resulting from open competition is resulting in greater demand for coherent, systems solutions for complex systems consisting of single or multiple vehicles operating in land, air and sea. These integrated systems are characterised by high capital value and extended life time, which together raise a requirement for evolution in system capability and the acceptance of change as an inherent characteristic of system infrastructure. The acceptance of change implies the need for a rigorous approach to the analysis and design of these systems which emphasises the achievement of modularity. In addition, since these systems often are required to operate in dynamic large, complex, uncertain, unstructured, non-benign environments without human intervention, there is a requirement for an intelligent adaptive ability which can react to environmental dynamics. Adaptive systems are more resistant to system and environmental changes potentially resulting in significant cost saving though increased operational life.

Benjamin Lussier et al in their paper “On Fault Tolerance and Robustness in Autonomous Systems” concerns about the dependability of autonomous systems, notably because of advanced decisional mechanisms and other artificial intelligence techniques on which the systems rely. The paper focuses on two non-exclusive approaches aiming to improve dependability, namely fault tolerance and robustness.

Kemin Zhou in his paper entitled "A New Approach to Robust and Fault Tolerant Control" summarizes a new approach to robust and fault control. This design approach is based on a variation of all controller parameterization in which two separate controllers are deployed: nominal performance controller and a robust controller. The controllers works in such a way that when a component (sensor, actuator, etc...) failure is detected, the controller structure is reconfigured by adding a robustness loop to compensate for the fault. This strategy works under various conditions. The price for achieving such a high performance and robust control is the complexity of the controller.

Ufuk Demirci and Feza Kerestecioglu in their paper entitled: "Fault Tolerant Control with Re-Configuring Sliding-Mode Schemes" presented a controller design method for linear MIMO systems in which a sliding mode controller is reconfigured in case of system faults. Faults are detected with the residual vector generated from a standard linear observer. Once a fault has been detected the fault distribution matrix can be obtained and used to update the corrective or equivalent control parts of the sliding mode controller. As a result, fault tolerant adaptive controllers keep the system performance within acceptable limits or at least avoid the system to wind-up. They have concluded that the second approach for reconfiguring sliding-mode controller which uses the extracted fault or disturbance information for the equivalent control part of the sliding-mode controller gives a better performance.

Balajee Kannan and Lynne E. Parker in their paper entitled, "Metrics for quantifying system performance in intelligent, fault-tolerant multi-robot teams" defined "fault-tolerant system" as any system that has the capability to diagnose and recover from faults. In their paper, they have outlined application independent metrics to measure fault-tolerance within the context of system performance. In addition, outlined potential methods to better interpret the obtained measures towards understanding the capabilities of the implemented system. Furthermore, a main focus of our approach is to capture the effect of intelligence, reasoning, or learning on the effective fault-tolerance of the system, rather than relying purely on traditional redundancy based measures. In this paper, they have presented an evaluation metric to measure the extent of fault-tolerance towards system improvement over a period of time. Furthermore, we evaluate two different multi-robot applications based on the defined metrics. Specifically, the research provides a quantitative measure for identifying system fault-tolerance in terms of efficiency, robustness and the extent of learning. In addition, this paper addressed the problem of developing application independent metrics for calculating the influence of fault tolerance towards system performance and identifies potential methods for analyzing the obtained measures towards evaluating the true capability of a multi-robot system.

Benjamin Lussier, et al have described in their paper entitled, "Fault Tolerance in Autonomous Systems: How and How Much?" In their paper, they address the execution of complex missions in uncertain environments and also address dependability as a whole, but focus specifically on fault tolerance. His paper presents several basic concepts:

1) Robustness and fault tolerance, both characterizing the resilience of a system towards particular adverse situations; robustness characterizes resilience towards uncertainties of the environment, and fault tolerance characterizes resilience towards faults affecting the system resources; and

2) Decisional mechanisms are central to autonomous systems as they embody the ability to dynamically select appropriate actions to achieve specific objectives; they are composed of knowledge specific to a domain of application and an inference mechanism used to solve problems.

They have summarized the state of the art in dependability and robustness mechanisms used in autonomous systems, and some conclusions that we drew from it. In particular:

1) Fault avoidance is largely privileged compared to other dependability means, although it rarely appears to be implemented intensively enough.

2) Development faults are hardly addressed by fault tolerant mechanisms in autonomous systems; robustness techniques somewhat compensate this problem, but are surely insufficient for critical applications.

Review of Diagnosis and Fault-Tolerant Control book by Blanke et. al covers several model-based failure detection techniques. Although failure detection is a well-defined challenge, the choice of system modelling technique can lead to quite different mathematical problems. A large portion of the book is devoted to presenting alternative modelling techniques. Chapter 1 introduces the fundamental problems of failure detection and fault tolerant control and presents an overview of the ideas behind the methods presented in later chapters. Chapter 1 also provides comprehensive definitions of terminology for use throughout the book. What is missing in this chapter (and in the rest of the book), however, are examples of applications of the methods covered. It would have been useful to give real-world examples that utilize the methods presented; for example, method A is used in the brake system of car X, a variation of method B is used in the autopilot of airplane Y, or method C will be used in a future space station. On the other hand, Chapter 2 presents two examples used throughout the book and, although these applications cannot be considered industrial, they are sufficiently complex to illustrate some problems with implementation of the methods. The first example concerns fluid-level control in a two tank system while the second example concerns steering control of a ship based on a simple model of the ship dynamics. A review of some dynamical system models is given in Chapter 3. In particular, state-space continuous time, discrete-time, discrete event, and hybrid systems are briefly discussed. Chapter 4 examines the application of some graphical analysis techniques to the study of fault propagation in component-based system architectures. The content of this chapter, however, seems unrelated to the rest of the book. Chapter 5 considers continuous time models and begins by deriving an interesting algorithm for finding redundancy among observed or known system variables that usually correspond to system inputs and outputs (u, y). The application of this algorithm to failure detection, however, is questionable because the redundancy relations involve first- and higher-order derivatives of u and y, information that cannot be obtained in many cases due to observation noise. The main results concerning failure detection techniques are presented in Chapter 6. This chapter reviews techniques based on analytical redundancy, in particular, the classical two-stage detection method for dynamical systems subject to additive noise. The stochastic setting with additive noise modelled as a stochastic white process, a two-stage detector was developed; the first stage consisted of a Kalman-based whitening filter generating a residual, and the second stage consisted of a canonical statistical test for deciding whether the residual had the expected statistical properties such as whiteness. Many variants and extensions of the two-stage approach have been studied in the literature and some of these extensions are presented in this chapter. In some cases, algorithms for implementing the techniques are also given and several examples are used to illustrate their application. Chapter 7 examines the fault-tolerant control problem. It is assumed that a finite number of faults is possible, and, for each fault, the dynamic behaviour of the system is modelled as a continuous-time dynamical system. Switching between systems is modelled by an automaton, and the notions of passive and active fault tolerant control are introduced. Passive control is closely related to classical robust control in that behavioural changes due to failure are viewed as model uncertainty and are taken into account in the design of the controller. Active control, on the other hand, is based on controller reconfiguration subsequent to failure detection and identification. Two techniques are investigated, the first being an optimal control approach, which is considered only in the simple case of linear system, state feedback, and quadratic cost. The proposed method does not seem realistic for applications. The second technique for fault-tolerant control is the virtual sensor-actuator. In the case of a sensor, the idea behind this technique is that, if a sensor fails, its output can be reconstructed from the other sensors, and the original controller can be used with the reconstructed observation. An analysis is presented for the limited case of static output feedback control. The applicability of this idea to the general dynamic feedback case is questionable; even stability properties do not seem to be preserved in the general case. The only general-purpose technique presented seems to be “the common sense approach,” which involves designing a controller for each model and switching the controller after a change of model has been detected. Chapter 8 explores a stochastic modelling technique for systems under surveillance. This technique uses a

stochastic automaton with the Markov property, which is usually called a controlled Markov chain. Surprisingly, the authors never refer to the literature on Markov chains or, in particular, to the literature on the application of Markov chains to failure detection. Some elementary properties of Markov chains are derived in Chapter 8, and applications to the failure detection problem are discussed. The idea of a virtual sensor is also examined in this context. In Chapter 9, the authors explore the supervision of a system where only quantized measurements of the inputs and outputs are available. This system is modelled as a stochastic automaton. Although this automaton does not have the Markov property, methods introduced in Chapter 8 are used for consideration of approximate models. There is no discussion, however, concerning stability. Finally, some of the methods discussed in earlier chapters are applied to examples introduced earlier.

Review of the paper of Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems by Xi-Cheng Lou, et. al, indicates that the robustness of the failure detection process consequently depends to a great degree on the reliability of the redundancy relations, which in turn is affected by the inevitable presence of model uncertainties. A wide variety of techniques has been proposed in recent years for the detection, isolation, and accommodation of failures in dynamic systems. In one way or another, all of these methods involve the generation of signals that are accentuated by the presence of particular failures if these failures have actually occurred. The procedures for generating these signals in – turn depend on models relating the measured variables. Consequently, if any errors in these models have effects on the observables that are at all like the effects of any of the failure modes, then these model errors may also accentuate the signals. This leads us directly to the issue of robust failure detection, that is, the design of a system that is maximally sensitive to the effects of failures and minimally sensitive to model errors. The authors indicate that all failure detection methods are based, either explicitly or implicitly, on the use of redundancy, i.e. on (possibly dynamic) relations among the measured variables. The paper addresses the problem of determining a redundancy relation that are optimally robust, in a sense that includes several major issues of importance in practical failure detection, and that provides a significant amount of intuition concerning the geometry of robust failure detection. The paper provides a procedure, involving the construction of a single matrix and its singular value decomposition, for the determination of a complete sequence of redundancy relations, ordered in terms of their level of robustness. This procedure also provides the basis for comparing levels of robustness in redundancy provided by different sets of sensors.

5.3 FAULT DETECTION AND ISOLATION

There are many methods used today to defeat fault. Fault diagnosis and fault tolerant control have become critically important in modern complex systems such as aircrafts. The basic concept of Fault Detection and Isolation is the ability of a system to diagnose the effect, cause, severity and nature of abnormal behaviour (i.e. faults and failures) in its components. The concept of Fault Tolerant Control is a closed-loop control system that tolerates component malfunctions while maintaining a desired degree of performance and stability. Figure 5.3 shows the possible location of faults that can enter into the control system.

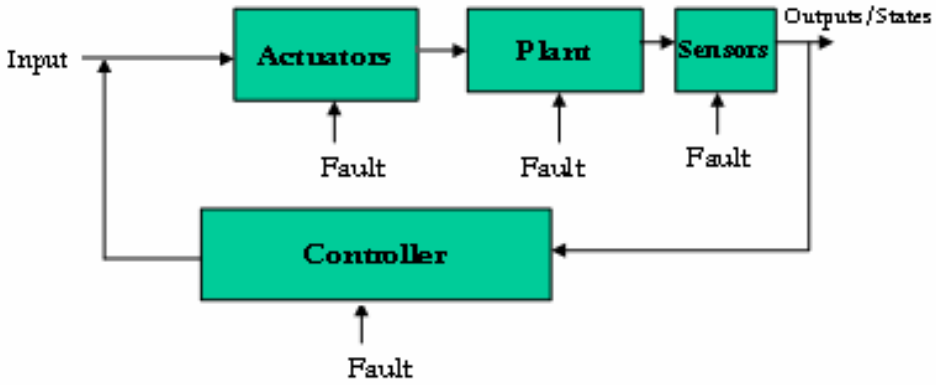


Figure 5.3: Location of Potential Fault in the Control System.

5.3.1 Review on Fault Tolerance

There are many techniques for Fault Tolerance. These techniques are: fault intolerance, fault detection and reconfiguration, and fault masking and reconfiguration. The first question comes to mind is: What is a Fault? Fault is different than error and failure. The definition of these is shown below:

> ***Fault****: An incorrect state of hardware or software resulting from failures of components, physical interference from the environment, operator error, or incorrect design.*
>
> ***Error:*** *The manifestation of a fault.*
>
> ***Failure:*** *A result of a delivered service deviating from the specified service caused by an error or fault.*

5.3.1.1 Fault Classifications

Based on Jaynarayan Lala and Richard Harper, fault can be classified various classification methods. This is illustrated in Table 5.3.

Table 5.3: Fault Classification

Classification Method	Types of Faults Involved
By Nature	Accidental Faults, Intentional Faults
By Origin	Physical Faults, Human-Made Faults, Internal/External, Design Faults, Operational Faults
By Persistence	Permanent Faults, Transient Faults, Intermittent Faults

Perhaps the best method to classify fault was suggested by D. P. Siewiorek and R. S. Swarz, in which the fault is based on it origin, or whether it is permanent in nature or not. This classification is shown in the following Figure 5.4.

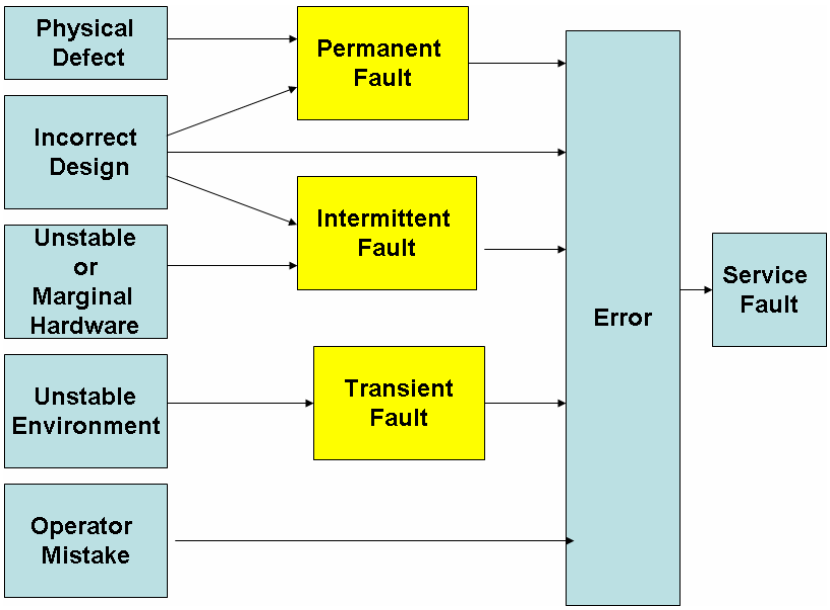


Figure 5.4: Fault Classifications.

5.3.1.2 How to Defeat Faults?

There are two distinct methods on how to defeat faults. These methods are:

1) ***Fault Intolerance/Prevention Methods***
2) ***Fault Tolerant Methods***
   - Redundancy
   - Fault Detection and Reconfiguration
   - Fault Masking
   - Software Fault Tolerance

Both methods are used for control system design today. There are various fault tolerance taxonomies. This is how Siewiorek and Swarz categorize fault tolerance techniques, as shown in Figure 5.5. There are other ways to categorize the methods, such as redundancy management systems. Fault tolerance is sometimes also called redundancy management. Mature systems all have ability to dynamically reconfigure after a fault is detected. That's true for fault detection systems as well as masking systems. (Redundancy Management).

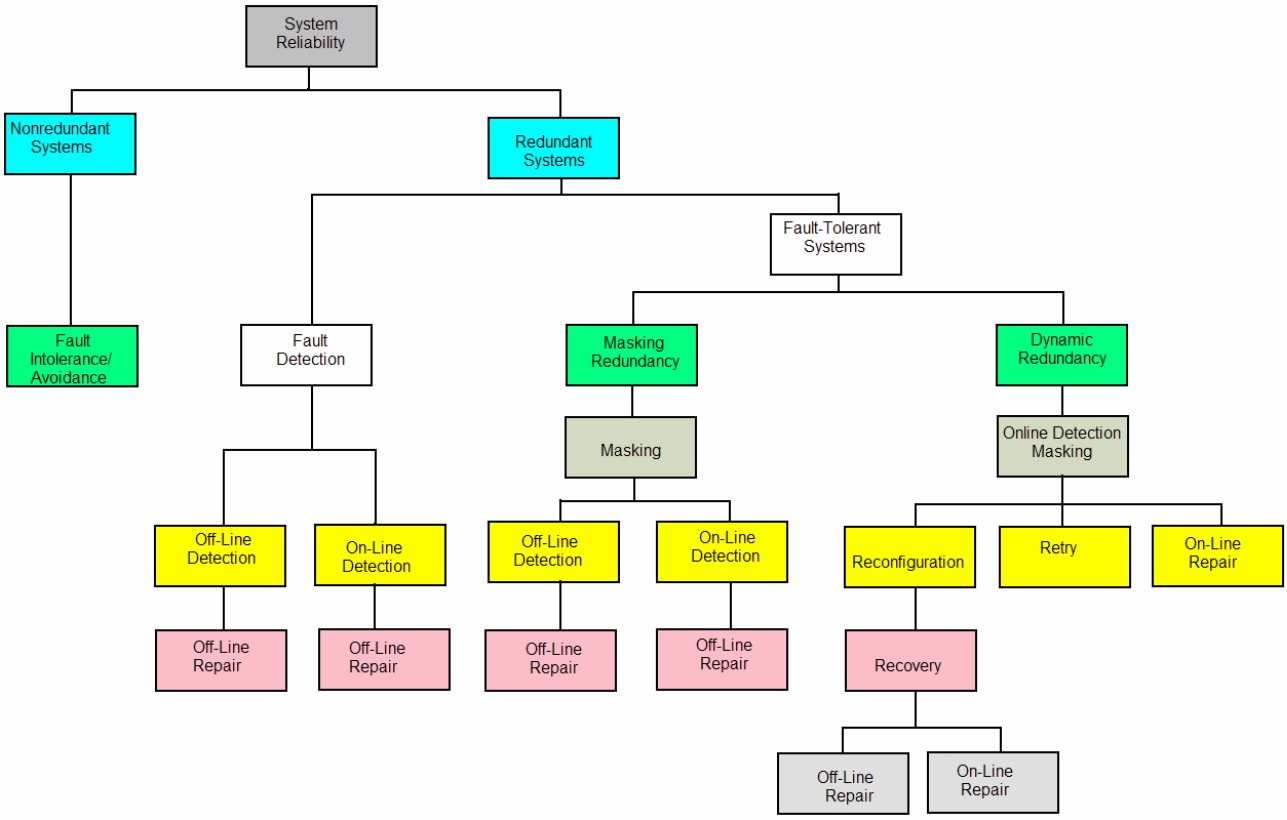


Figure 5.5: Taxonomy of System-Failure Response Strategies.

In a paper by Patton, a graphical presentation of Fault Detection and Identification (now Isolation) (FDI) for a robust control and reconfigurable control is shown in Figure 5.6. Here, the scattered areas of research are shown towards a robust, fault-tolerant system of the future.

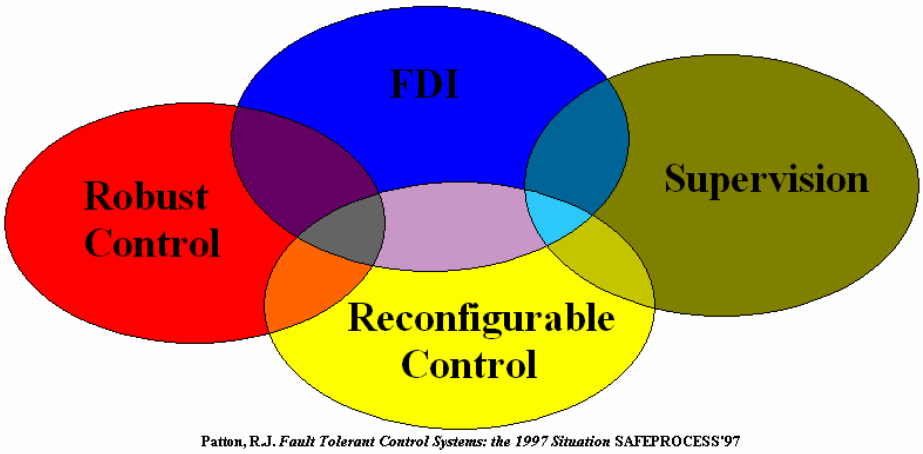


Figure 5.6: Graphical Presentation of Fault Detection and Identification.

Since no system in the real world can work perfectly all the time under all conditions, it is critical to be able to detect and identify the possible faults in the system as early as possible so that measures can be taken to prevent significant performance degradation or damages to the system. In the last twenty some

years, fault diagnosis of dynamic systems has received much attention and significant progress has been made in searching for model-based diagnosis techniques. Many techniques have been developed for fault detection and fault tolerant control. However, the issue of robustness of fault detection and fault tolerant control has not been sufficiently addressed. The conceptual of model-based diagnostics is shown in Figure 5.7 [Jie Chen and R.J. Patton]. Model-based diagnosis employs analytic redundancy compare actual components with an analytic model (mathematical model). It depends on the validity of the model, and the ability to accurately model a system, and it is relatively straight forward for linear systems, but difficult for nonlinear systems (most software-based systems).

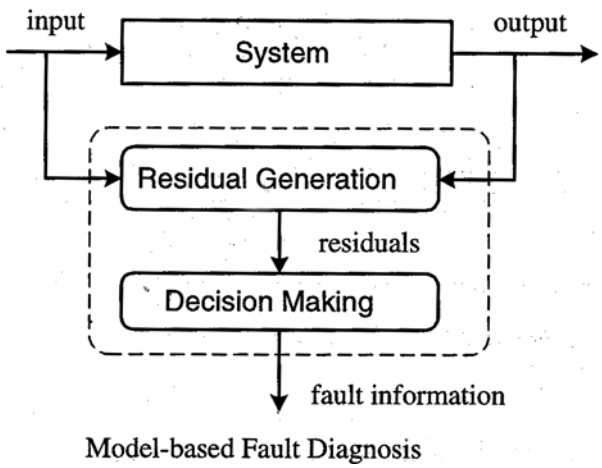


Figure 5.7: Conceptual Structure of Model-Based Fault Diagnostics.

Since disturbances, noise, and model uncertainties are unavoidable for any practical systems, it is essential in the design of any fault diagnosis/fault tolerant control system to take these effects into consideration so that fault diagnosis/tolerant control can be done reliably and robustly. Fault tolerance can be implemented to improve on one hand safety, and on the other hand reliability towards incorrect or incomplete knowledge due to system deficiencies, design compromise for efficiency, and faults in the decision procedure. Robustness can be implemented to improve reliability towards unexpected changes of situation due to the environment or system dynamics, and incorrect or incomplete knowledge due to lack of observability. Figure 5.8 [Jie Chen and R.J. Patton] illustrates the general schematics of Model-based fault detection technique.

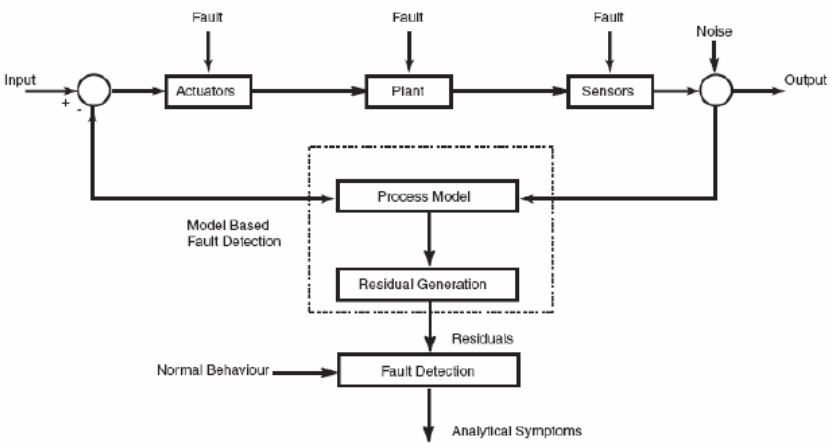


Figure 5.8: General Scheme of Model-Based Fault Detection.

Youmin Zhang's lecture – FTC part one has suggested an active fault tolerant control which is depicted in Figure 5.9. As shown, the fault, disturbances, and noise can enter at any point. The key point is how to defeat faults. This was shown in Section 5.2.1.

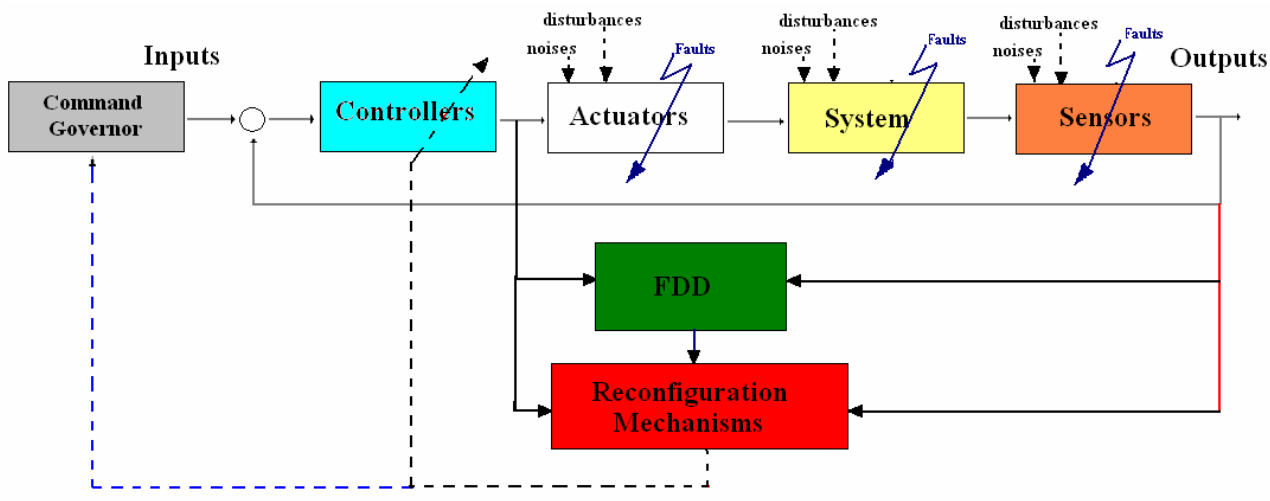


Figure 5.9: One Scheme of Active Fault Tolerant Control.

Rune M. Jensen and Manuela M. Veloso and Randal E. Bryant on their paper entitled, "Fault Tolerant Planning: Toward Probabilistic Uncertainty Models in Symbolic Non-Deterministic Planning" have a goal of a more probabilistic uncertainty model by distinguishing between likely primary effects and unlikely secondary effects of actions. We consider the practically important case where secondary effects are failures, and introduce n-fault tolerant plans that are robust for up to n faults occurring during plan execution. Fault tolerant plans are more restrictive than weak plans, but more relaxed than strong cyclic and strong plans. In their paper, they take a first step in this direction by introducing a new class of fault tolerant non-deterministic plans. Their work was motivated by two observations:

1) Non-determinism in real-world domains is often caused by infrequent errors that make otherwise deterministic actions fail; and

2) Normally, no actions are guaranteed to succeed. They have concluded that Fault tolerant planning is a first step toward more refined models of uncertainty in Symbolic Non-Deterministic Planning (SNDP). A fruitful direction for future work is to move further in this direction and consider fault tolerant plans that are adjusted to the likelihood of faults or to consider probabilistic solution classes with other transition semantics than faults.

5.4 FAULT AND FAILURE ACCOMMODATION

Future military land, air and sea based vehicles will require advanced system concepts that enable new vehicle capabilities under both nominal and adverse conditions. These systems will be required to maintain control under faults and failures, prevent and recover from vehicle loss of control conditions, and provide automatic collision avoidance capability. A fault is defined as a "malfunction" of any physical component or a sub-system that results in its failure to perform as designed. A system fault can arise from natural wear and tear of mechanical or electrical components, external unknown catastrophic disturbances, and improper maintenance of electro-mechanical components. It is highly desirable that when a fault occurs, it is detected as soon as possible necessary action can be taken. This timely response to faults reduces any disastrous consequences. For this reason, fault detection and isolation (FDI) methods are becoming very desirable for advanced vehicle systems. Faults being dynamic in nature,

the reconfiguration method should be capable of accommodating them quickly, especially for complex systems like advanced military aircrafts. Reconfiguration based on adaptive techniques demands a fast detection and isolation of a fault and is computationally involved. The systems of the future will have a reconfigurable flight control system to a transport aircraft model for the purpose of achieving an integrated failure accommodation and upset recovery system. The major task of such systems is the process of detection, identification and accommodation of faults and failures (FDIA). A number of approaches exist, of which model-based techniques show particular promise. Model-based approaches use analytical redundancy to generate residuals for the aircraft parameters that can be used to indicate the occurrence of a fault or failure. Actions such as switching between redundant components or modifying control laws can then be taken to accommodate the fault.

A review of "Robust Fault-Tolerant Control for Aircraft Systems" Thesis by Phalguna Kumar Rachinayani has addressed the need to design controllers that guarantee both stability and performance upon the occurrence of faults. He has presented different methodologies to design robust controllers that guarantee both stability and robustness for actuator faults and uncertainties. In the first part of this thesis, he has introduced the classical uncertainty formulation using Linear Fractional Transformation (LFT) and describes LFT's special cases-norm bounded and convex polytopic uncertainty descriptions. Practical methods to formulate these uncertainty structures are described. In the same spirit, formulation of faults and their modelling for robust control system design is provided. In the second part of this thesis, he has demonstrated the application of a Luenberger observer for fast Fault Diagnosis and Isolation (FDI). He has described the methodology to design a robust optimal control for actuator faults and present controller reconfiguration mechanism based on switching for the design of Fault Tolerant Control (FTC). System with both norm bounded uncertainties and actuator faults is formulated and an analytic method to find a robust stabilizing and guaranteed cost reliable controllers are also mentioned.

5.4.1 Robustness and Redundancy

Traditional approaches to fault tolerance – reliability and redundancy. Why? Robustness – ability to continue to function within acceptable bounds in the presence of uncertainties.

Various techniques have been proposed in recent years for the detection, isolation, and accommodation of failures in dynamic systems. All of these techniques involve the generation of signals by sensors that are accentuated by the presence of particular failures, if these failures have actually been occurred. These sensor signals in turn depend on models relating the measured variables. Consequently, if any errors in these models have effects on the observables that are at all like the effects of any of the failure modes, then these model errors may also accentuate the signals. This leads to the issue of robust failure detection, that is, the design of a system that is maximally sensitive to the effects of failures and minimally sensitive to model errors. As a result, designing a failure detection system that is insensitive to model errors (rather than designing a system that attempts to compensate the detection algorithm must be concentrated by estimating on-line uncertainties.

Application robustness, defined as the ability to provide a contained degradation in performance when the algorithm solving the application is perturbed in its structural parameters, has an immediate impact on the design of a reliable system. The robustness of the failure detection process consequently depends to a great degree on the reliability of the redundancy relations, which in turn is affected by the inevitable presence of model uncertainties.

5.4.2 Fault-Tolerant Control and Reconfiguration

Fault tolerance is one of the principle mechanisms for achieving high reliability and high availability in propulsion and any other systems. System controls should be equipped with appropriate fault-tolerance schemes to ensure their reliable and safe operation in the presence of component failures. As an example,

Figure 5.10 describes the block diagram structure of a fault tolerant control system [Phalguna Kumar Rachinayani].

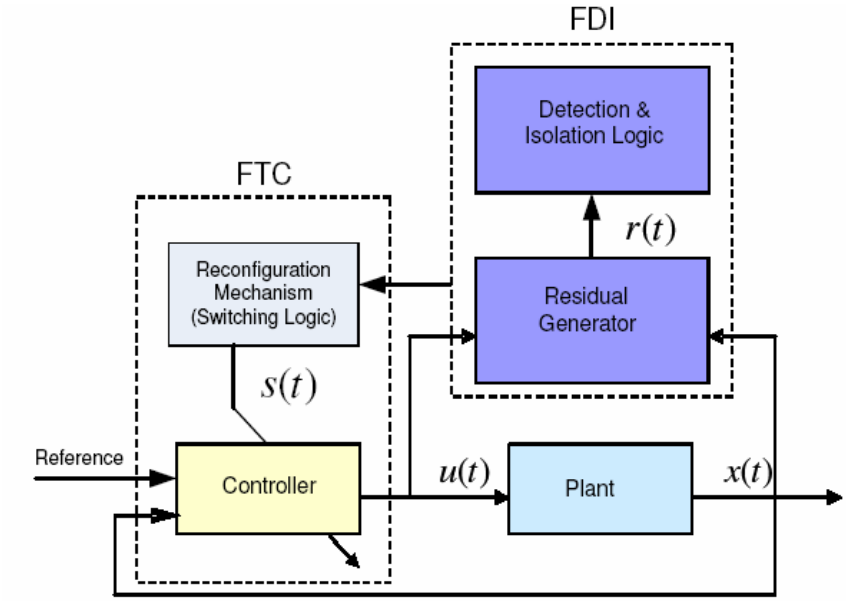


Figure 5.10: Block Structure of Reconfigurable Fault Tolerant Control System.

System reconfiguration, which enhances reliability by dynamically using spatial redundancy, is generally the most time-consuming fault-/error- handling stage. The reconfiguration latency, defined as the time taken for reconfiguring a system upon failure detection or mode change. Various fault reconfiguration strategies to locate and isolate the failures have been proposed. These schemes allow the controller to operate with a degraded performance even in the presence of faults. Fault-tolerant approach for discrete-event systems can be achieved by integrating sensor fusion within a diagnosis and control reconfiguration mechanism. This approach can automatically compute fault-tolerant control/reconfiguration actions which maintain pre-specified control objectives. Reconfiguration is based on model-based diagnostic representations and algorithms, and integrates diagnostics and control reconfiguration for discrete event systems using a single modelling mechanism and suite of control algorithms. Given failures in the system, a modelling approach can facilitates diagnostic isolation while performing sensor fusion to minimize false alarms, thereby increasing tolerance to sensor faults as well as (non-measurable) component faults. The algorithm works dynamically and individually: system reconfiguration starts instantly upon the emergence of a fault and the replacement of a new faulty component is considered independently from previous replacements. Hybrid Estimation and Control is becoming very popular. A broad 'Hybrid' philosophy is needed; such as 'hybrid modelling' which takes the benefits of both analytical and empirical models; 'hybrid system dynamics' which incorporate both the continuous system dynamics and discrete event dynamics which will be more realistic in real engine operations and finally 'hybrid algorithms' which take advantage of both analytical and intelligent (rule based) algorithms to tackle these complex issues. Such effective integration of analytical and intelligent tools towards the development of new and innovative identification/control strategies for complex dynamic systems is of paramount importance in achieving a reconfiguration scheme, including a reconfiguration algorithm.

5.5 PROGNOSTICS

Traditionally, maintenance on aircraft has been conducted on-time or on-failure. An example of the former is the overhaul or replacement of aircraft engines after a stated number of operating hours. In avionics and

flight-control systems, the latter is more common, and redundant components are provided in order to preclude loss of function following a single component failure for flight- or safety-critical functions. The desire to lower maintenance costs and associated system down time has led toward driving component failure rates in all aircraft subsystems sufficiently low to allow more maintenance to be conducted on failure or specified degradation of components rather than on time. However, unpredicted failures or the concern for such failures limits the degree to which maintenance and logistics footprint (and associated costs) can be minimized. Only if it were possible to predict exactly when a component (by serial number) would fail or when its performance would fall below that allowable would it be possible to further drive these costs down. By the mid 1990s, it appeared the technology (in the form of electronics, sensors, and computational technology in highly miniature form) existed or was close enough to realization to postulate tracking of performance on an item by item basis to predict exactly when it would fail or cease to perform acceptably, and do so with enough warning to have a replacement part standing by at the right place and time to minimize maintenance impact and footprint (and safely get the maximum life out of each component). Thus, instead of *diagnosing* and correcting problems only after they occurred, the aircraft could proactively predict impending problems in sufficient time to preclude their occurrence. This concept is called *Prognostics*.

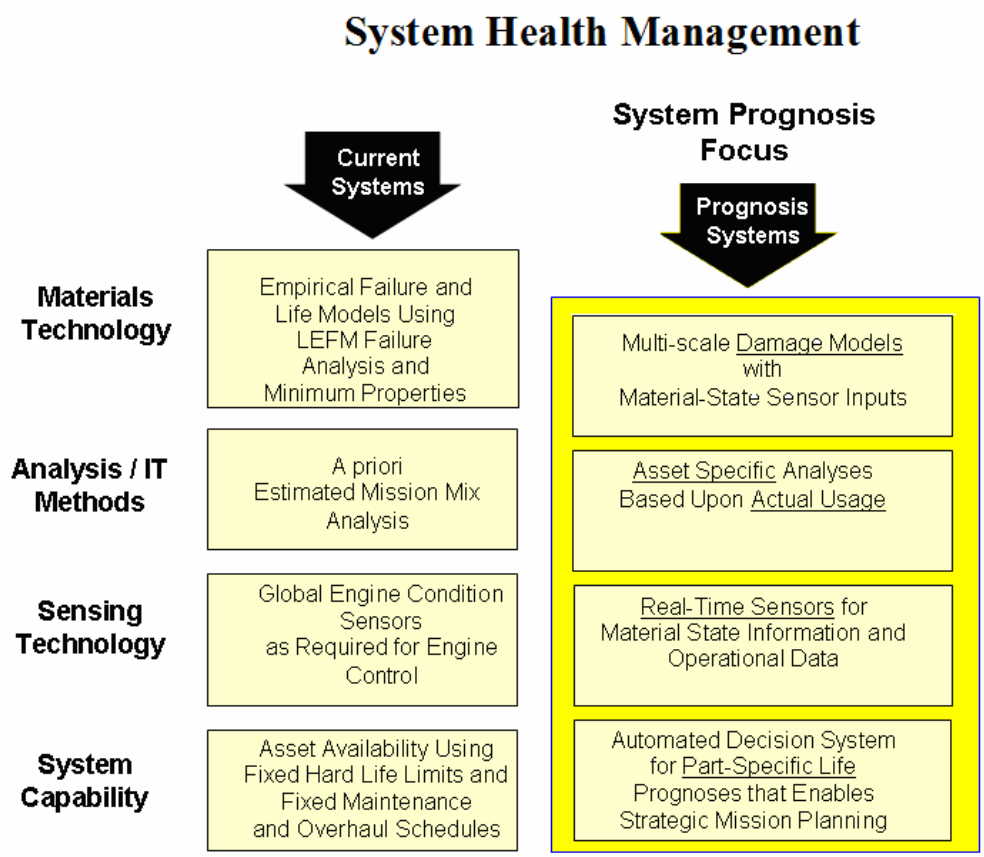


Figure 5.11: System Health Management.

5.5.1 Prognostic Assessment

There are a number of initiatives being pursued to enhance the capabilities of the present maintenance system. There are many programs that are in the process developing prognostic technologies. Many of these technologies are being matured so that they can be deployed on the systems as soon as possible to try to eliminate some of the shortcomings of the present maintenance process. Specifically, propulsion system

developers have taken the lead in the development and deployment of prognostic systems that will greatly enhance the present maintenance system and provide more availability of those systems. In order to accomplish this, the exploitation of what is called the "prognostic region" must be accomplished. This is a departure from past experience that used Built-in Test (BIT) for diagnostics. BIT utilized a percentage above a threshold value to determine when a failure had occurred. At this value the system was determined to be failed and needed to be repaired by replacement parts or other means to get the system functioning again. The paradigm shift has been to develop prognostic algorithms that will predict when a failure will occur and determine the life remaining to prevent a hard failure. This paradigm change in maintenance philosophy will eliminate the reactive maintenance event that might take place during a period of critical operations when the systems are needed to perform. The establishment of the BIT threshold was already established by the diagnostic engineer in the past when determining the BIT threshold; it was just a determination as to where that threshold should be and when to set a Fault Isolation Code (FIC). In determining the prognostic region, it is the same process except we want to move up the curve to determine how much operating time is left until we get a functional or hard failure. It is the slope of the curve that we are attempting to calculate so that an accurate Time to Failure (TTF) can be estimated. If an accurate assessment of the slope of the curve can be derived, then scheduling of the repair can be accomplished prior to a functional failure. While this will not improve reliability of the system, the reliability is inherent to the developed system; it will greatly enhance the availability of the system. The algorithms to predict the remaining life of the system prior to failure are under development.

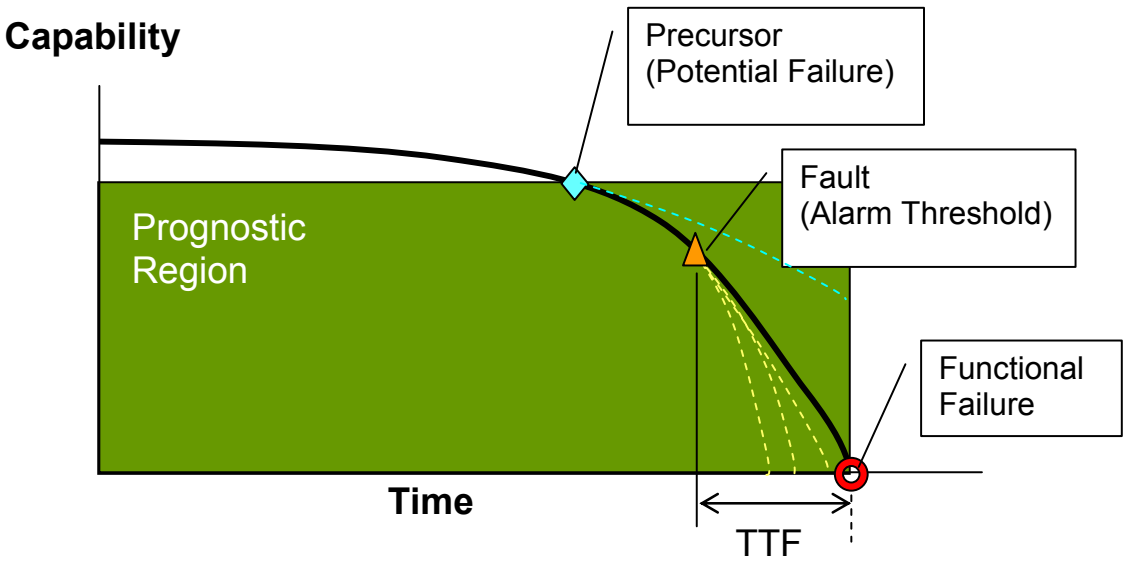


Figure 5.12: Prognostic Regions.

5.5.2 Development of Diagnostic Systems as a Precursor for Prognostics

There have been many developments in Prognostics that have evolved from advanced diagnostic techniques. Initially, a prognostic was thought of as an advanced diagnostic technique, but in reality, it is a paradigm shift of a new way of managing the life remaining and determining the health of the system at any point in time. Previous diagnostic developments were aimed at improving the diagnostic capability of systems and eliminating Cannot Duplicate (CND) and ReTest OK (RTOK) type problems. The development of the Flight Control Maintenance Diagnostic System (FCMDS) was to look at the diagnostic accuracy in terms of the RTOK rate. The RTOK occurs if a Line Replaceable Unit (LRU) is removed from an aircraft during aircraft troubleshooting, but no fault can be found during bench testing or subjecting the unit to an end-to-end testing on a test system. This places a great demand on the supply system and will require the stocking of many spares to keep up with the flight line demand for spare parts

and LRUs. One way around this is to put the LRU into another aircraft to see if it will function normally and work in that aircraft. This was tried at one time, but using a multi million dollar aircraft as a test bed becomes very expensive and very time consuming and this was not a successful maintenance concept. It was discovered that taking the LRU out of the system could potentially remove or change some of the diagnostic evidence and trying to diagnose systems with parts removed would not yield successful results. The FCMDS program was then structured around troubleshooting the system with all LRUs in place. A software model of the flight control system was developed using model based techniques so that anything that could happen in the on board flight hardware could be put in the model to diagnose the system. This was one of the first instantiations of model based reasoning. Additionally, it was realized that the wring system itself cause many problems to occur, and many fault codes to set when a signal that did not get to its destination. While the setting of the fault code might have pointed the technician to a certain LRU, it was really a case of the signal not getting to the LRU for it to provide an appropriate response to the stimulation. Upon realization of this, the wiring diagnostic module developed as an integral part of the diagnostic system.

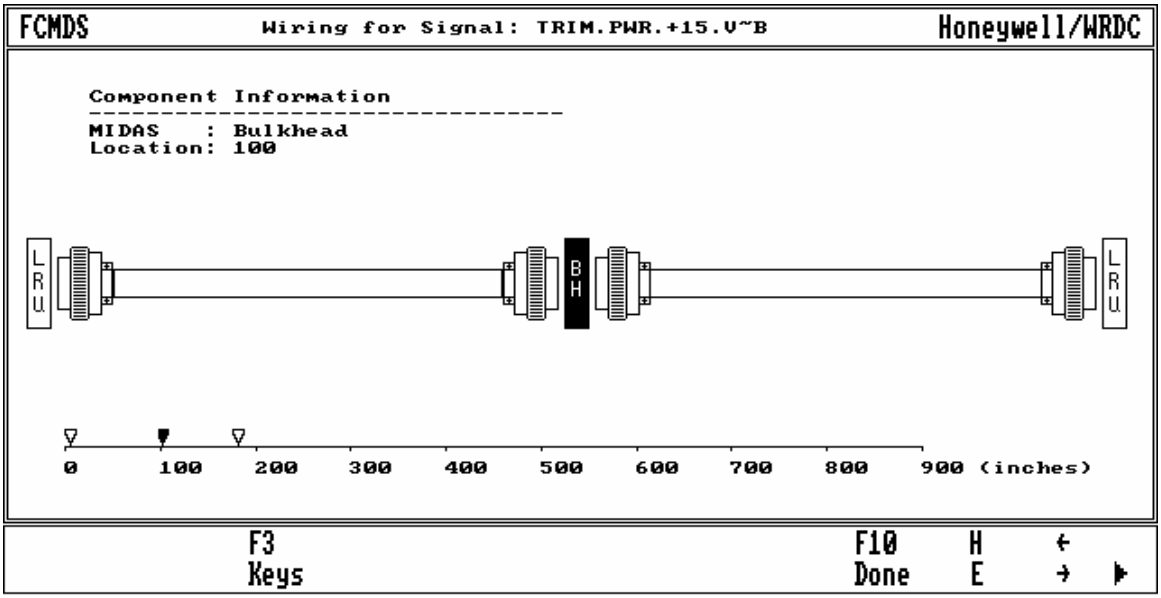


Figure 5.13: Wiring Diagnostic Module.

Wiring is an integral part of the control system because it connects the parts of the system. In a sense, the wiring is the neural network of the system. Wiring is often neglected in development of fault isolation manuals and techniques because wiring failures typically to not occur until the aircraft has been in service for at least five years. Unfortunately, when a wring failure does occur, the information necessary to rectify the problem may be missing or in a form that is difficult to obtain. In addition, there are few technicians that are expert in diagnosing wiring problems, and when a wiring diagnostician is needed they are very hard to find. Diagnosing wiring is more of an art than a science, and it is something that comes with practice and diligence and a mental attitude of the ability to find the problem. The FCMDS made wiring an integral part of the diagnostic approach. Since information flow was contained in the diagnostic model, the wires were modelled as well. Therefore, each diagnostic observation that was mapped into the model includes all the wiring connections implicitly. The wiring diagnostic model developed for the FCMDS system was a manually tested with something like a DVOM for resolution of the problem. Since that development, automated techniques are under developments that are nearing maturity. Embedding an automated wiring diagnostic module within the system takes the human out of the loop and finds faults during system operations taking into account many factors that can not be tested when the aircraft is on the

ground. Attributes such as air speed, g loading, altitude, temperature, angle of attack, can all have an impact on the wiring system. When the aircraft returns from a flight, the evidence of a failure that occurred during g loading in a turn is not obtainable on the ground, therefore this problem is almost impossible to diagnose. With the development of embedded wiring diagnostic techniques collecting this data during a flight gives the technician a much better chance of finding the problem the first time it occurs.

When diagnostic information became accessible through a 1553 bus interface, the technician's job got harder. Suddenly to diagnose an aircraft failure, the technician must be able to read hexadecimal code in the digital flight control memory. Many pieces of data must be obtained and interpreted, a source for clerical errors. People make errors when performing tasks like these. FCMDS through an electronic interface drew information from the aircraft in digital form and performed the necessary interpretation of the hexadecimal codes. Computers are well suited to this type of clerical task both in terms of speed of acquisition and accuracy of results. More important, however, is that the data collection became a fundamental aspect of the FCMDS test language design. In a test, if the information is required that the system can obtain automatically, the diagnostic system will establish communications with the aircraft data bus and acquire the information. The technician is informed through the dialog display when and what data is accessed and including the status of the system.

The FCMDS development represented a precursor to the development of prognostics. In FCMDS, the human controlled the diagnostic process and did the entire test sequence manually, guided by the system after inputting test results from individual tests. Taking into account that the system was designed for diagnostics and used a BIT system, the addition of FCMDS to the diagnostic process proved very successful. The field test proved that the FCMDS system could significantly reduce false removals by 80%, and reduce the average diagnostic time by 25% and greatly enhance the level of performance of the maintenance technician using FCMDS over currently used methods at that point in time. This also provided the catalyst to support a two level maintenance concept that was under development at the time. The decrease in the RTOK rate would provide the additional spares to support a two level maintenance concept.

Following the development of FCMDS, a means of automating many of the manual processes that were in place at that time of the FCMDS implementation have been computerized. Rather than having the technician analyze all the data, the data is collected and sent to a central server where data mining techniques are used to analyze all the relevant data from the system. Analysis techniques are embedded into the central server to look at the trends taking place in the system over time. These trends will allow the system to make projections over time as to when the system will need maintenance prior to a predicted functional failure. The system will alert the maintenance operations centre as to the repair schedule for the effected system. This new concept is called the predictive maintenance concept. The prognostic attributes of the system are used to make predictions as to when the system will fail. It is the prognostic attributes that are being developed that will make the predictive maintenance concept possible. The next big step in this process is to integrate the current research in prognostic development is to integrate this with the logistics infrastructure.

The definition of data fusion has developed over ten years. Initially in 1987, the JDL Data Fusion Subgroup described data fusion as a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, complete and timely assessments of situations and threats, and their significance. Additionally, the data fusion process was characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results. More recently, in 1998, Steinberg simplified the definition for data fusion as a process of combining data or information to estimate or predict entity states. Spanning this time frame, many DoD projects under the guise of "sensor integration" have accomplished the "fusing" of data from several

sensors and information sources. Through its evolution, sensor integration has been further developed and termed both "Sensor Fusion", "Data Fusion", and "Information Fusion" without a clear definition of the distinctions between them. Capabilities developed were predominantly focused on on-board weapon systems sensor capabilities ranging from target tracking, weapons control, and system navigation. Disciplines, methods, and techniques emerging from sensor fusion investigations run the gamut and include general statistical methods, hypothesis testing, graph theory, data representation, resource management, artificial intelligence (AI), fuzzy logic, and neutral networks. Evolving form the earlier projects, military systems have solved more ambitious tasks and have applied more types of sensors and information sources.

The DoD challenge now is to move forward in data management of the sensor data collected from these on-board weapon systems and to fuse this operational data with data generated through test, inspection and repair of the system, as well as manually reported operational data. It is believed that integrating these diverse data sets will enable advanced capabilities for logistics support, to include maintenance practices and processes. A research area currently being developed within the Air Force Research Laboratory (AFRL) calling for this paradigm in data integration is Integrated Systems Health Management (ISHM). The AFRL ISHM program concept is a fully systems-of-systems vision. The AFRL projects which fall under its umbrella range from sensor development, life prediction methodologies, to asset management and advance to improved system engineering design processes. Key capabilities for ISHM is the DoD vision for Conditioned Based Maintenance Plus (CBM+) to improve upon knowledge-based tools and applications.

AFRL's Health Management research links to the Air Force Transformation Flight Plan, which addressed development of advanced Condition-Based-Maintenance-Plus (CBM+) capabilities (AF/XPXC, 2003). CBM+ is the maintenance management policy being adopted from the Department of Defence (DoD) strategic vision titled "Future Logistics Enterprise" ((FLE), now referred to as: Force-centric Logistics Enterprise). The DoD FLE vision documents were paramount in initiating the force transformation efforts being implemented by US military services today. The FLE is comprised of six focus areas, one of which is CBM+, [McRuer, D.D. Graham, E. Krendel, and W. Reisner, Jr.].

The CBM+ maintenance management concept is unique to the DoD. As documented in the DoD FLE, the interim policy states:

> "Condition-based-maintenance (CBM) can be defined as a set of maintenance processes and capabilities derived, in large part, from real-time assessment of weapon system condition obtained from embedded sensors and/or external tests and measurements using portable equipment. The goal of CBM is to perform maintenance only upon evidence of need. CBM+ expands on the basic concepts of Condition-Based Maintenance by encompassing other technologies, processes, and procedures that enable improved maintenance and logistics processes", [McRuer, D.T. and E.S. Krendel].

Each military service is in the process of adopting the DoD FLE guidance on CBM+, making further refinements upon the definition [Tustin, A]. The FLE goal is to implement CBM+ with enabling technologies that advance and implement the capabilities or programs, listed in Figure 5.14 below, on future, acquisition, and sustaining weapon systems where possible.

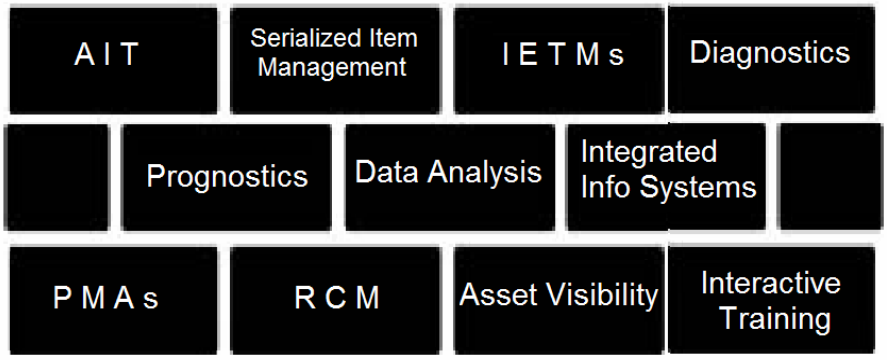


Figure 5.14: DoD CBM+ Initiative: Listed CBM+ Capabilities or Programs Being Targeted under the DoD's FLE.

5.5.3 US Air Force CBM+

The Air Force kicked-off its examination for developing its CBM+ policy by enlisting the Air Force Logistics Management Agency (AFLMA) to conduct a comprehensive study on the subject. The AFLMA study concentrated on "Air Force CBM+ implementation as a basis for establishing an Air Force CBM+ policy", [McRuer, D.T., and E.S. Krendel]. This study was completed in September 2003.

Systems Health Management is not specifically called out in the AFLMA study. However, their definition for CBM+ combines On-Condition Maintenance attributes with the Reliability Centered Maintenance, Joint Total Asset Visibility, and System Health Management concepts. The AFLMA study proposed the Air Force adopt the DoD FLE CBM+ definition with the following additional phrase: **"*These future and existing technologies, processes and procedures will be addressed during the capabilities planning, acquisition, sustaining and disposal of a weapon system*"**. The AFLMA also proposed the following technologies should be listed under the AF CBM+ as enablers:

- Prognostics
- Diagnostics
- Data analysis
- Interactive Training
- Portable Maintenance Aids
- Integrated Information Systems
- Automatic Identification Technology
- Interactive Electronic Technical Manuals

New on-board and off-board technological capabilities are needed to support transforming maintenance to implement CBM+. The overarching AFRL vision for Health Management fully integrates CBM+ technologies, with research primarily focusing on prognostics and diagnostics capabilities that influence or drive capabilities of the other technologies listed.

AFRL's Health Management research is in line with the Air Forces' desire to develop transformational capabilities as called out in the Air Force Transformation Flight Plan, published November 2003. In the Flight Plan's chapter titled "Developing Transformational Capabilities", the first of sixteen goals stated is to develop "Seamless joint machine-to-machine integration of all manned, unmanned, and space systems". In this statement, "machine–to-machine" can be interpreted to include air-to-air, ground-to-ground, air-to-

ground and vice versa, the word "all" would imply systems in both sustainment and new acquisitions, encompassing the spectrum of logistics and operational functional applications.

Further discussion in the transformational capabilities chapter, under Section F. titled "Agile Combat Support", is focused on transforming logistics. This section calls out several Information Technology (IT) programs, as well as Condition Based Maintenance-plus, as key programs or future systems enabling transformational capabilities to meet the lighter, leaner, and faster combat support goals being addressed here. AFRL is working to bring these capabilities to the Air Force, which requires collaboration with a diverse systems infrastructure.

5.5.4 Integration with the Logistics Infrastructure

In order to take advantage of the advances in developed prognostic systems, these prognostic algorithms must be proven and tested in an operational environment. Additionally, the prognostic algorithms must be integrated with the present and future logistics environment. At the present time there have been many prognostic developments that have been advertised and some that have been tested, but not many in an operational environment and none with the integrated with the logistics infrastructure. While there is still a great deal of work to do in development of prognostics to determine the life remaining of systems and components, it is time to start integrating this technology with the logistics infrastructure. The Air Force is moving from many stand alone IT systems that provide single purpose assessments for systems in the aircraft to a centralized IT system (i.e. propulsion oil debris data) that will provide the needed services for the Air Force. This system, formerly known as Air Force Knowledge Systems (AFKS) is called Global Command Support System (GCSS). As many diagnosticians have known for many years that having the data to analyze in a central location makes the job much easier. Propulsion systems, for example, have many diverse data systems to support operations and maintenance, and most are stand alone and some are not available to be stored in a permanent location. This data is lost and will never be recovered. A movement is underway to integrate all the data into a central location so that is possible to do "predictive maintenance" on propulsion systems. The idea behind predictive maintenance is to use the prognostic TTF to estimate the time remaining. Using the multiple data sets from the propulsion system and development and use of data mining techniques will allow a greater resolution into the prediction of the life remaining until a functional failure occurs. The resolution of that aspect can be used by the present logistic infrastructure to start to schedule propulsion repairs as "future maintenance events" that occur prior to a functional failure. With GCSS, the supply data, operational schedule, personnel and proper tools can be scheduled for a future maintenance event. There is a great deal of research that will be needed to accomplish this futuristic maintenance system for the Air Force.

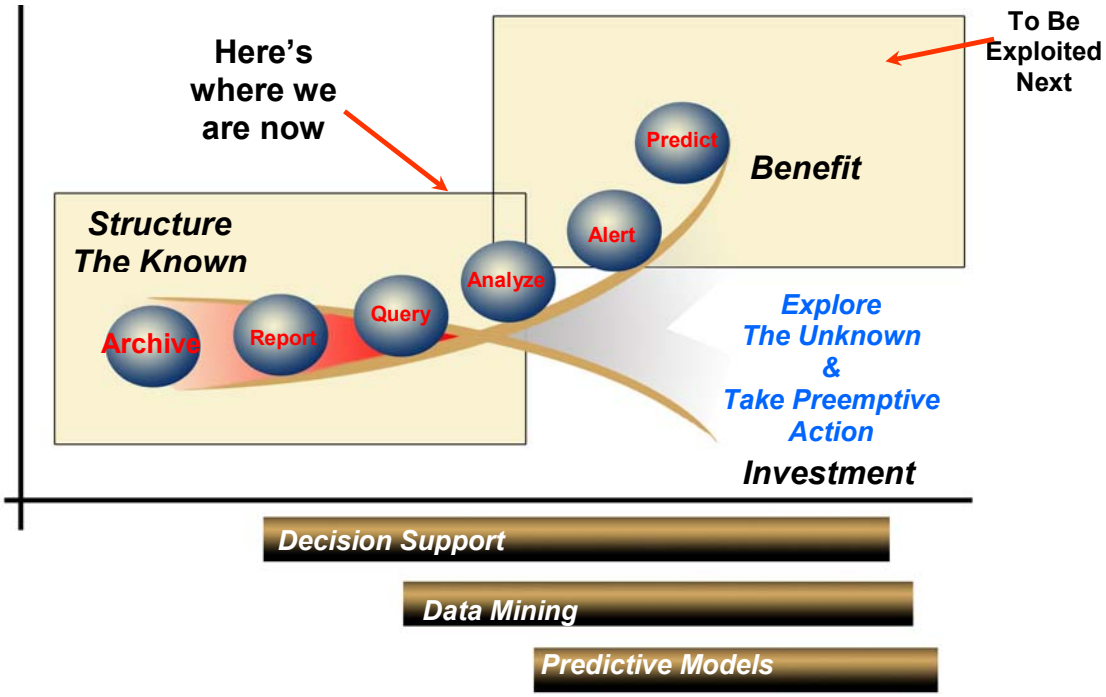


Figure 5.15: GCSS AF Data Services Capability Concept.

5.6 INFORMATION FUSION

Information fusion methods in principle have been applied ever since data has to be handled. There is no task finished with receiving data, they always have to be related to other data or previous knowledge to make sense and enable decisions. A very good example for basic, yet difficult information fusion is stereoscopic vision: one image alone just offers two-dimensional information, but fused with another, very similar image information enhances to 3D. Because information fusion is such a basic technique it can be found in many disciplines, from biology to the design of computational databases. Although there are a lot of solutions for various applications, no theoretic or conceptual work on information fusion has been done a long time. Even the term "information fusion" has not been existed up to 1991, when the JDL (Joint Directors of Laboratories, DoD, USA) introduced a definition meant for military applications. Due to increasing complexity and new applications groups of all disciplines became aware of the need for systematic research on that area. Now there are journals, web-sites and conferences dealing with information fusion as an independent yet interdisciplinary topic.

There is no general information fusion method which works for all addressed problems. In fact basic information fusion techniques of different disciplines have nothing in common except to provide a better use of data, which may be unstructured, inaccurate, overwhelming by volume or difficult to interpret. Often information fusion is referred as "filtering", "pre-processing", "decision making", and so on. A short overview on the different approaches shows the variety and bandwidth of information fusion.

Information fusion subdivides in two major categories: Condensation and interpretation of data. Under "condensation" all methods are summarized, that shrink huge amounts of redundant data to valid information, that combine complementary data to new insights or that find hidden information in disordered data. A lot of well established methods like state estimators (Kalman filter, Luenberger observer, maximum likelihood methods and others), averaging concepts and voting schemes have been designed for that purpose. More recent methods are (active) perception management or entropy methods, which both are used to get the most useful information out of sensors or other data. Interpretation of data

is less algorithmic and more "smart" than the condensation task. Here data is associated, sorted, classified and decisions are derived. This is accomplished by expert systems, artificial intelligence and the often used probabilistic methods like Bayesian Networks or Dempster-Shafer theory. This discipline is object of ambitious development. In Figure 5.16 the main methods are outlined.

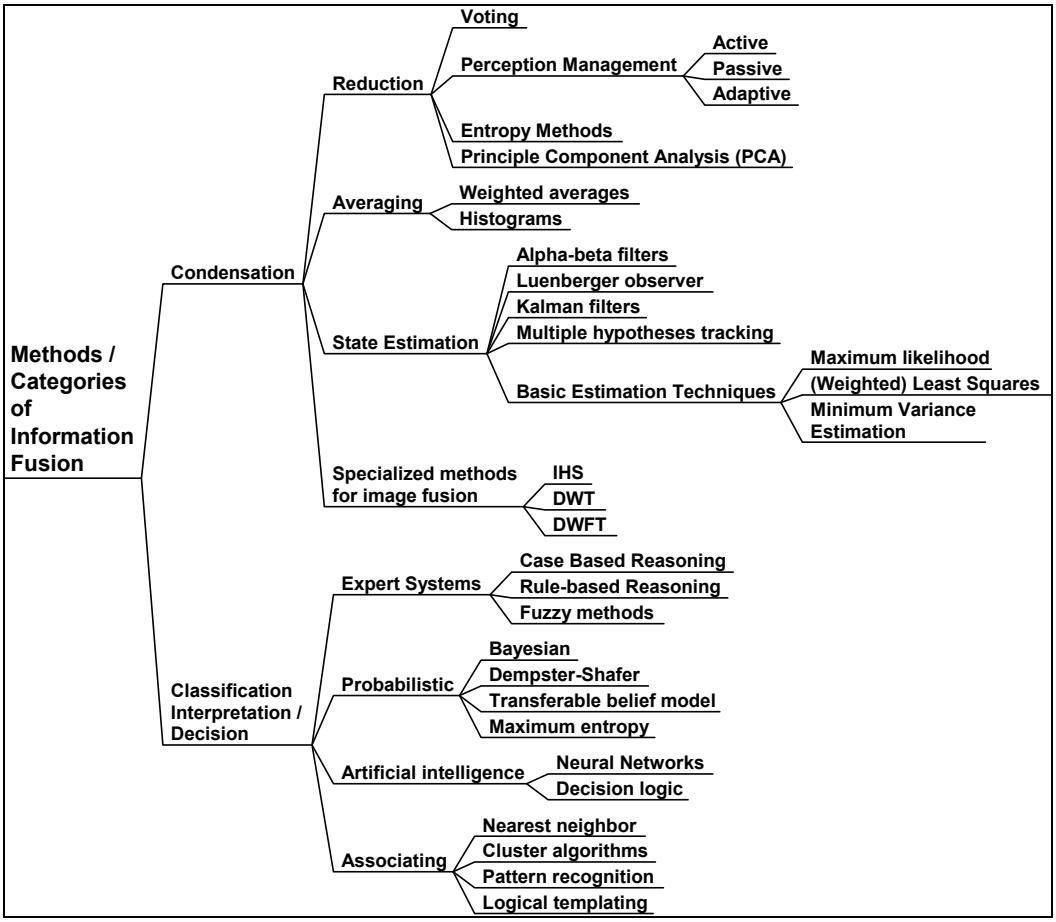


Figure 5.16: Methods of Information Fusion.

Technically it can also be decided between different types data fusion:

- Fusion of temporally data of **one** sensor.
- Fusion of data of multiple sensors of the **same type**.
- Fusion of multiple sensors of **different type**.
- Fusion of sensors and models.
- Fusion of sensors and a-priory knowledge.
- Fusion at raw-data, feature and decision level.

5.6.1 Definition of the Fusion Process

The following is the information fusion process:

- The combining or merging of data from different sources in a manner that provides extra information and/or better quality than any of the single sources involved (see Figure 5.17).

- The process of acquisition, filtering, correlation and integration of relevant information from various sources (see Figure 5.18).

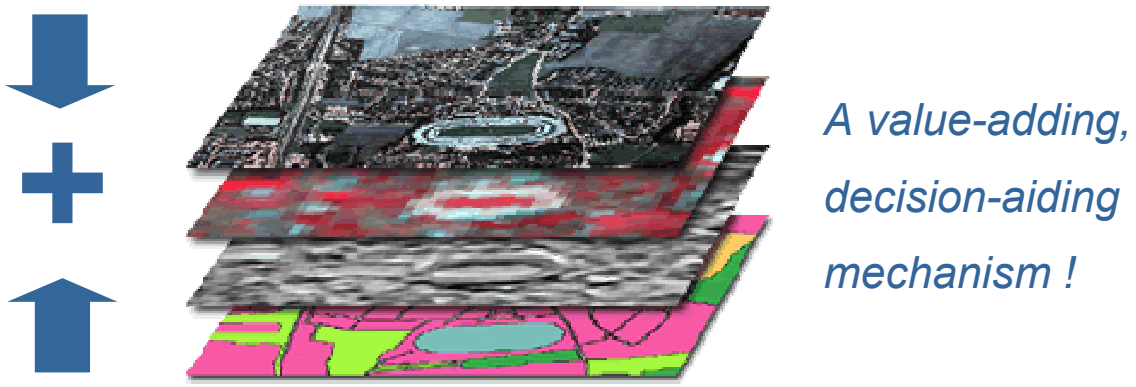


Figure 5.17: Information Fusion Levels.

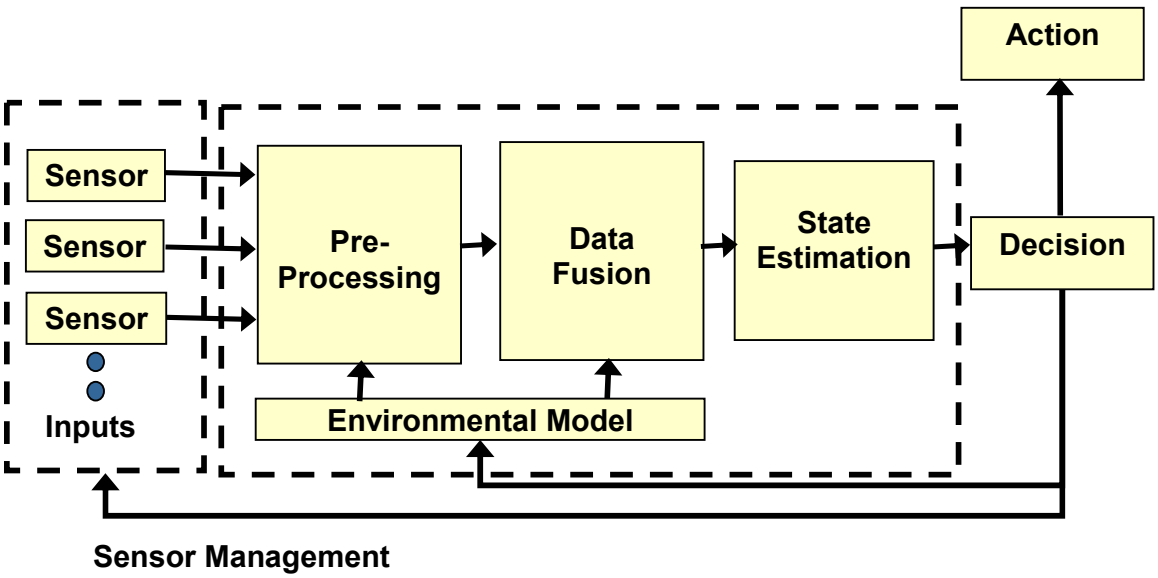


Figure 5.18: Acquisition Data Process.

5.6.2 Data Types

Data sets typically have various, recognisable patterns of distribution of the individual values. The following describes the data types encountered in various types of system operation:

- Not all of the data will be in a nice, user-friendly, easy-to-use form.
- Military information is often defined in terms of:
  - IMINT (Imagery – FLIR, Visible, SAR, etc.);
  - ELINT (Electronic Emissions – RADAR, MAWS);
  - HUMINT (Human Generated Intelligence);

  - SIGINT (Signals); and
  - COMINT (Communications).

- Combining different types:
  - Imagery (e.g. IR and Visible versions of the same scene) can be difficult if the two sensors are not co-located; and
  - How does one combine an image with text (HUMINT) that may, or may not, describe the same scene?

5.6.3 Categories of Fusion Concepts

Data fusion is defined as a formal framework in which are expressed the means and tools for the alliance of data originating from different sources. Data fusion techniques combine data from multiple sensors, and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application. In exploring the concept, data fusion has the following elements:

- Condensation:
  - Shrink redundant data to valid information;
  - Combine complementary data to new insights; and
  - Find hidden information in disordered data.

- Classification / Interpretation. Data is:
  - Associated;
  - Sorted;
  - Classified; and
  - Decisions are derived.

- Complementary Fusion:
  - E.g. several visual sensors pointing in different directions.

- Competitive Fusion:
  - E.g. measuring of a distance: laser ranger sensor, acoustic (ultrasonic) sensor pointed on the same point.

- Cooperative Fusion:
  - E.g. fusion of physical measurements
    2D images → 3D representation.

5.6.4 Data Acquisition Problems

Within the current environment, the typical approach taken for AFRL health management programs is to acquire sample data from the multiple information systems used in supporting maintenance on the chosen legacy system, to accomplish development of the programs proof-of-concept. One would think that the targeted legacy weapon system would also benefit from this research as well, but this is not often the case

because the IT systems supporting sustaining operations are many and diverse. So, in the early stage of health management research, data acquisition is one of the primary issues AFRL managers contend with.

Each support function of a legacy weapon system typically utilizes multiple Air Force standard, and weapon system unique stove-pipe or stand-alone IT applications for support equipment. Typically these IT systems only share data if required by stake holders within the functional domain of the support activity, as depicted in Figure 5.19. Examples of the functional domain data for propulsion are: oil analysis results, digital diagnostic tool results, or on-board operational performance data.

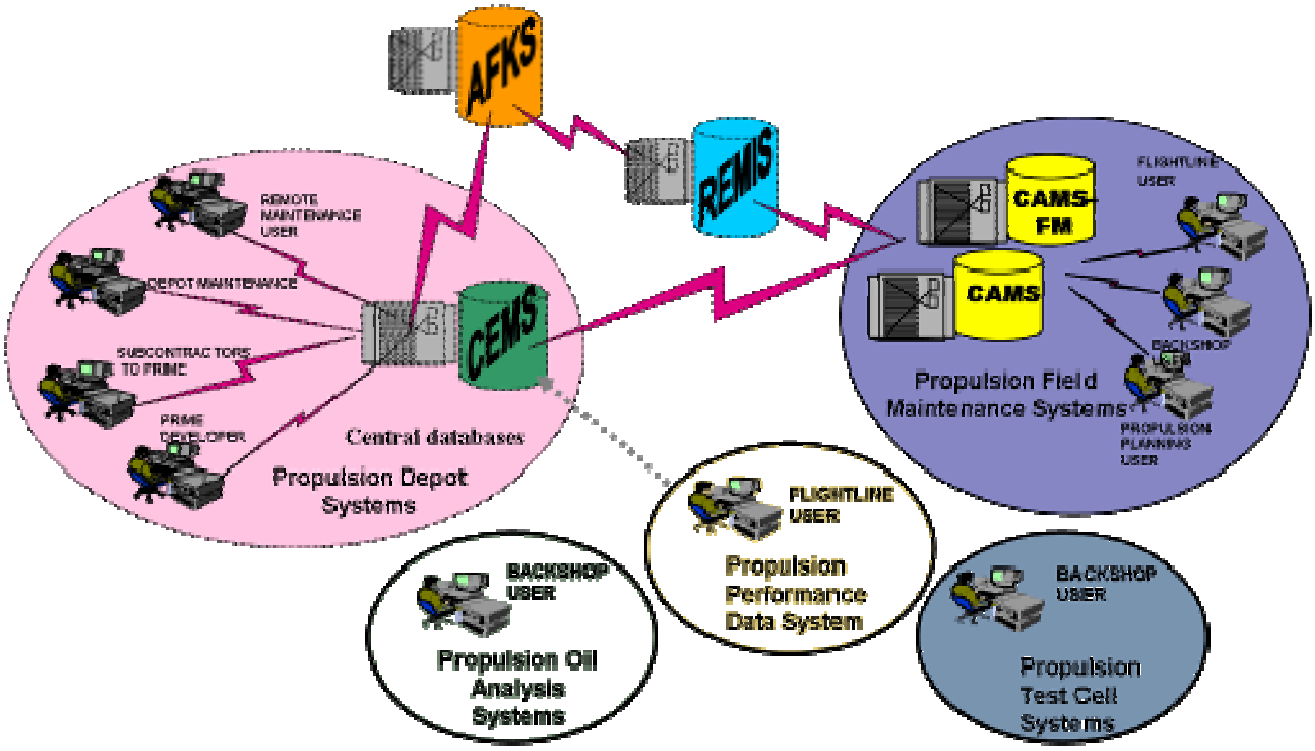


Figure 5.19: Legacy Propulsion IT System Interfaces: Initial GCSS Implementation of Legacy Central Data Systems Data. This figure also depicts legacy central data systems interfaces with base level systems and peripheral systems with no interface requirements that support propulsion assets.

It seems baffling that a research community, belonging to an organization as large as the Air Force, doesn't have readily available the operational data needed to conduct research. However, the AFRL engineer will typically spend 80% of the project time acquiring data and the remaining 20% on analysis. Therefore, the data situation requires the AFRL project engineer to spend countless hours coordinating with IT owners, or system users of the data sources, identified to support the specific program initiative.

The data acquisition problem is further compounded when a research project calls for data from multiple information sources, which is typically the norm for application concepts that hinge on data integration or data fusion. Also, there are times when multiple AFRL programs require the same data set from one of the available information systems, which under current practices, cause duplicate efforts from researchers for acquiring data to occur. The EHM team's goal is to reverse the two percentages previously stated, so that 80% of the time is dedicated to analyzing the data and 20% to acquiring the data. To overcome the data acquisition obstacle, the AFRL EHM team is seeking to forge a new paradigm in IT program management through collaboration, beginning with the GCSS Program.

5.6.5 Data Management Policy

Air Force data management policy has historically concentrated on the manual processes required to report equipment availability and maintenance activity. Any guidance on the disposition of data collected digitally, via test equipment or from equipment operation, has been left to the managing system/equipment program office or a Major Command to provide. Therefore, the collection of digitally generated data is sporadic and it's not normally stored or transferred for purposes beyond the test or operation being conducted. For this reason, the AFRL EHM team has initiated discussion among AFRL ISHM team members and Air Force policy makers on the need to draft an overarching data management policy to be supported by standard IT systems.

New policy is required to expand the central collection of system performance data and other data generated from sophisticated electronic test equipment. Such policy would provide the possibility of integrating on-board weapon systems collected data with other functionally related data. Thus enabling the creation of advanced CBM+ capabilities, or "the integrated application of a collection of advanced engineering, maintenance and information technologies to improve maintenance and logistics practices" as proposed by the AFLMA.

5.6.6 AFRL GCSS Research Environment

The goal is to establish an AFRL research environment where both programs (AFRL and GCSS) managers and their IT developers will collaborate on delivering program initiatives.

Over the life of a project, current thinking is that phase one would consist of concept and methodology development, which should include data acquisition requirements, data relationships and draft interface specification documentation. During phase two, concept maturation and demonstration, the team would concentrate on data acquisition and the database architecture required for developing the research capability.

Finally phase three, technology transition, would complete maturation of the Engine Health Management (EHM) decision support concept by broadening the capability to address all pertinent components within the targeted propulsion system. The theory behind this program progression concept is that the capability would be developed in a relevant environment or in an IT environment capable of supporting the concept. For the team this reality transforms for AFRL, GCSS, and their collective customer by meeting the program goals for a predictive maintenance system.

Changes to AFRL EHM program management have already been implemented using this program development strategy concept. The EHM team has begun targeting additional applicable IT acquisition programs open to this collaboration concept. The team is currently managing a number of Small Business Research projects to start the implementation of this strategy.

This conceptual process and strategic alliance is being developed by propulsion engineers and logisticians to bridge IT capability development gaps, which have existed for many years. Upon implementation of this concept the ability to use GCSS to store and acquire data will be realized. The team believes a strategic alliance with GCSS will ensure the probability of success in developing workable prognostic and diagnostic capabilities.

The EHM team also foresees a need to demonstrate new capability concepts that could be applied to sustainment logistics IT systems. They believe similar arrangements for AFRL program development could be implemented to bridge CBM+ capability gaps as legacy IT systems transform. However, much work is needed in changing paradigms on both fronts of the systems engineering process between the S&T and legacy IT development organizations.

There have been many developments in the past that would have gone forward faster if there had been partnering relationships and/or a GCSS system in which to conduct research. Finally, the EHM team likens the use of a GCSS research environment to the use of their Demonstrator Engine program, which is used to satisfy testing requirements of S&T propulsion component programs and is critical to program success. It is envisioned that an AFRL Research Environment within GCSS will become a reality in the very near future.

Finally, once the strategic alliance or program partnering concept is approved and implemented, this arrangement will benefit a broad range of other AFRL Science and Technology (S&T) initiatives, related to data fusion, modeling concepts, or the development of knowledge-based e-tool capabilities. Although the alliance team is currently focusing on the propulsion labs Engine Health Management S&T need and use of GCSS resources, the GCSS Architect expects the final program partnering document will serve as a "GCSS User Self-Development Guide" model, for even a broader range of Air Force customers to use in developing their desired knowledge-based capabilities.

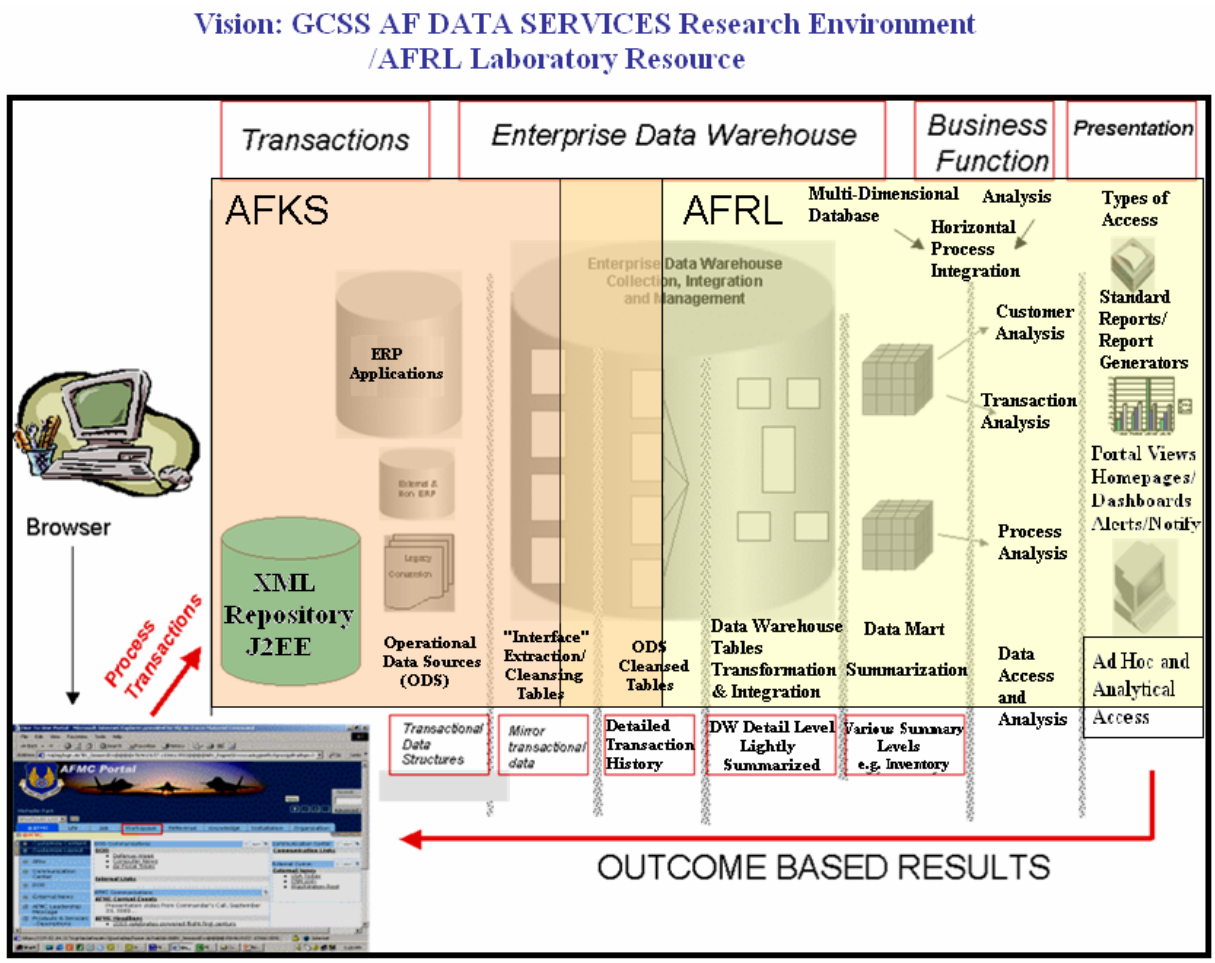


Figure 5.20: Integrated Vehicle Health Management (IVHM) and Condition Based Maintenance (CBM).

5.7 APPLICATIONS IN HUMAN-MACHINE INTERFACE

Application and usability of human-machine interface (HMI) aspects of tools/systems designed to enhance human functioning is vital to the augmentation of human-machine partnerships in the military

environment. In general terms, HMI systems are designed and optimized to visualize, control and report on data that is either being polled from or sent from other systems in near-real-time (see Figure 5.21). HMI are pilot-centered systems that provide prioritized data at the right time and in the right format to optimize situational awareness and increase mission effectiveness. Some aspects of HMI are listed below:

- Multimodal Interfaces:
  - Refers to interfaces that support non-GUI interfaces
  - e.g. complementary speech and pen input
  - Often distributed on a network

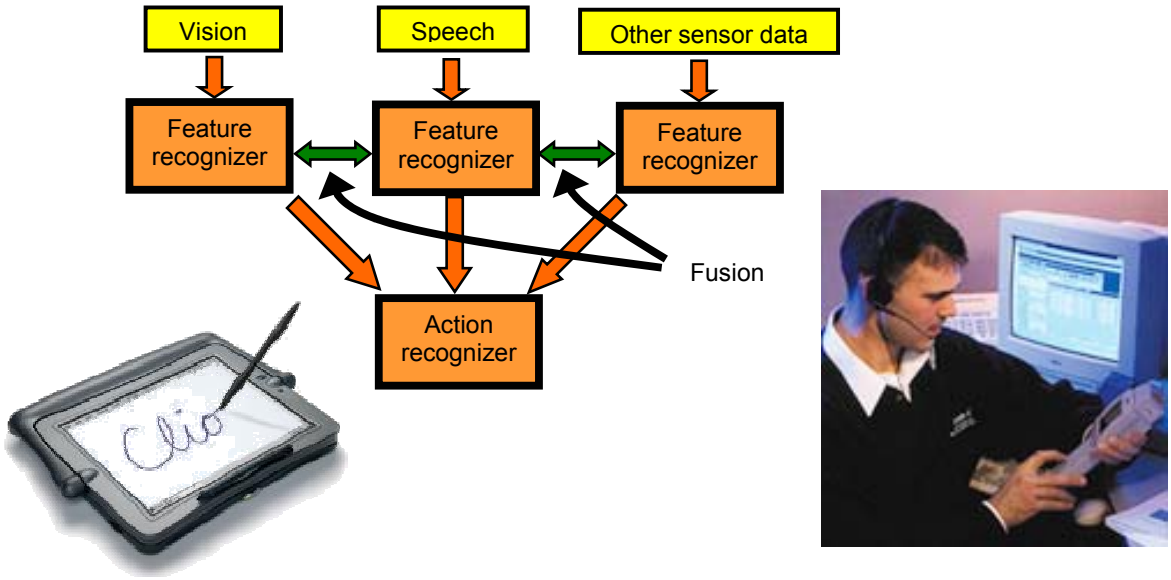


Figure 5.21: Human-Machine Interface.

5.8 SYSTEM HEALTH MANAGEMENT

Increased demands on autonomy in high performance aircrafts, UAVs and spacecrafts has led to the development of the concepts of Integrated Vehicle Health Management (IVHM) [Tustin, A.] and Condition Based Maintenance (CBM) [Tustin, A.]. Health monitoring or health assessment is an integral part of IVHM and CBM systems.

Health monitoring, as the name implies, involves monitoring the state or condition of components in a system. Most of the algorithms for health monitoring of aerospace components have evolved from fault tolerant control design applications. In these applications, in addition to monitoring the health of components, appropriate reconfiguration actions have to be initiated based on the diagnostic/prognostic information.

Efficient HM could be achieved only by combining data/information from multiple sources. Thus, data fusion is an important component of health monitoring systems where correlated data from multiple sources are used to determine the state of a system. Data fusion helps in achieving robustness, extended spatial and temporal coverage, increased confidence, reduced ambiguity, lower uncertainty and improved resolution [McRuer, D. T., D. Graham and E. Krendek]. Data fusion also helps in increasing diagnostic visibility, reliability and in reducing the number of false alarms due to the effective utilization of all the available data/information. Data gathered from different types of sensors need to be combined with

linguistic, knowledge-based information for achieving efficient monitoring. In systems where the health of several components is to be monitored, choice of appropriate data fusion architecture plays an important role. Also for applications like engine HM, a single diagnostic system may not be adequate to cover all types of faults/failures that could occur. In such cases, methods for combining information from several diagnostic systems to make appropriate decisions are also to be given due consideration.

The common goal of IVHM [Tustin, A.] and CBM [Tustin, A.] systems is to achieve automated health management. Hence, several common features would be evident in the description of both the systems given below.

The purpose of IVHM is to achieve the ability to perform vehicle maintenance based on component/subsystem condition and operational requirements, to automate flight certification, monitor and manage/schedule maintenance resources. For unknown fault/anomaly detection and decision support, IVHM utilizes model based diagnostics, intelligent data collection, signal processing, context models and correlation techniques.

In general, IVHM involves:

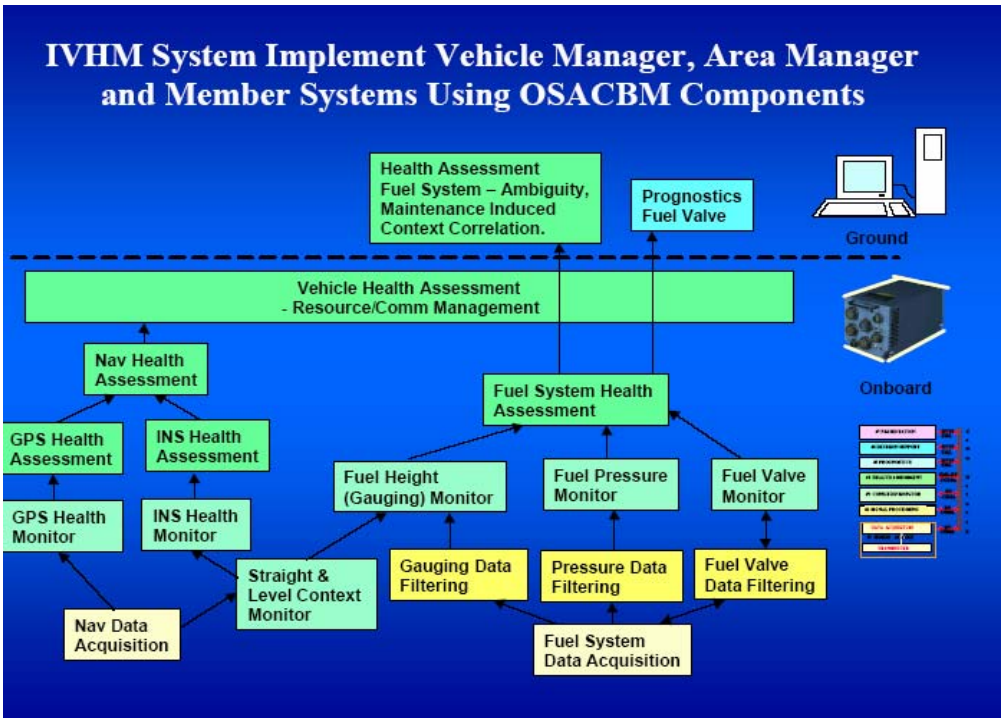
i) Diagnostics – to determine the state or capability of a component to perform its function(s);

ii) Prognostics – predictive diagnostics to determine the remaining life or time span of proper operation of a component;

iii) Health Monitoring – to monitor the state or condition of a component; and

iv) Health Management to make appropriate decisions about maintenance actions based on diagnostics/prognostics information, available resources and operational demand.

Figure 5.22 [Tustin, A.] shows the components of an IVHM system for an aircraft. Using the data gathered from the GPS, INS and fuel systems, aircraft health assessment is made using OSACBM (Open system architecture for Condition based maintenance) components [Tustin, A.]. Fusion of system identification and component health information helps in advanced diagnostics and prognostics for adaptive flight control reconfiguration as also shown in the example in Figure 5.22.



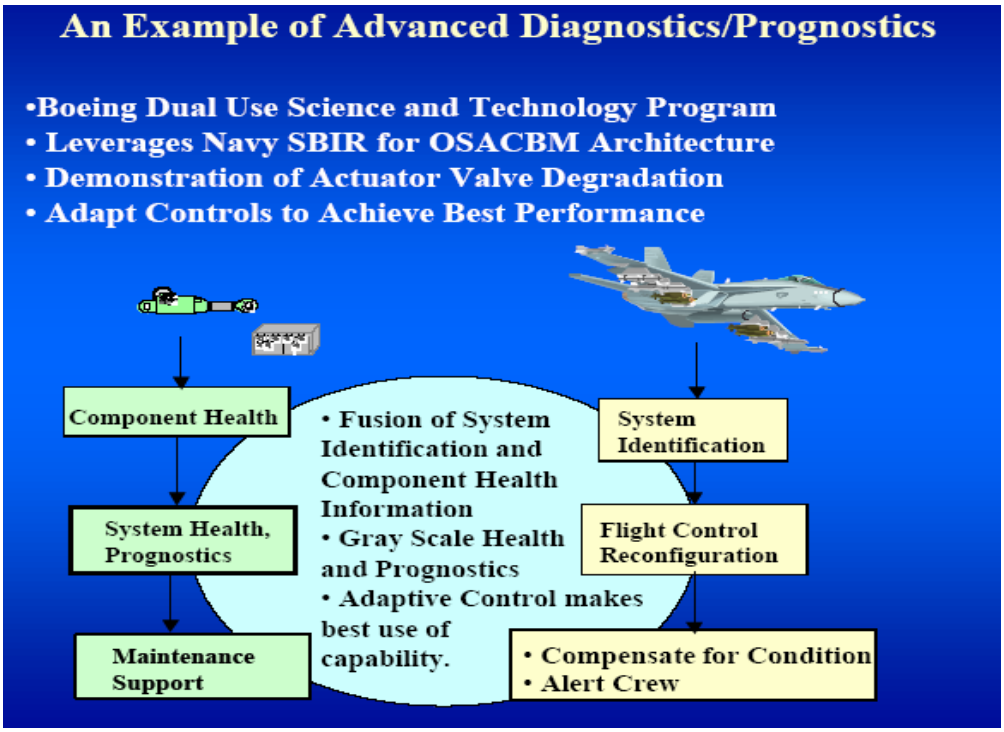


Figure 5.22: [Tustin, A.]: IVHM System and Advanced Diagnostics/Prognostics.

Condition based maintenance (CBM) and prognostics is necessary in military as well as industrial applications [Tustin, A.]. Since the CBM involves the integration of a variety of hardware and software components, a standard to encompass the entire range of functions from data collection to recommendation of specific maintenance actions is being evolved by Boeing, Rockwell and Penn State Applied Research Laboratory, ARL [Tustin, A.]. Figure 5.23 [McRuer, D. D. Graham, and E. Krendel]

shows the system overview for CBM system based on the OSA-CBM concept for health monitoring of an electro hydraulic spoiler actuator. The various elements in the figure are described briefly below.

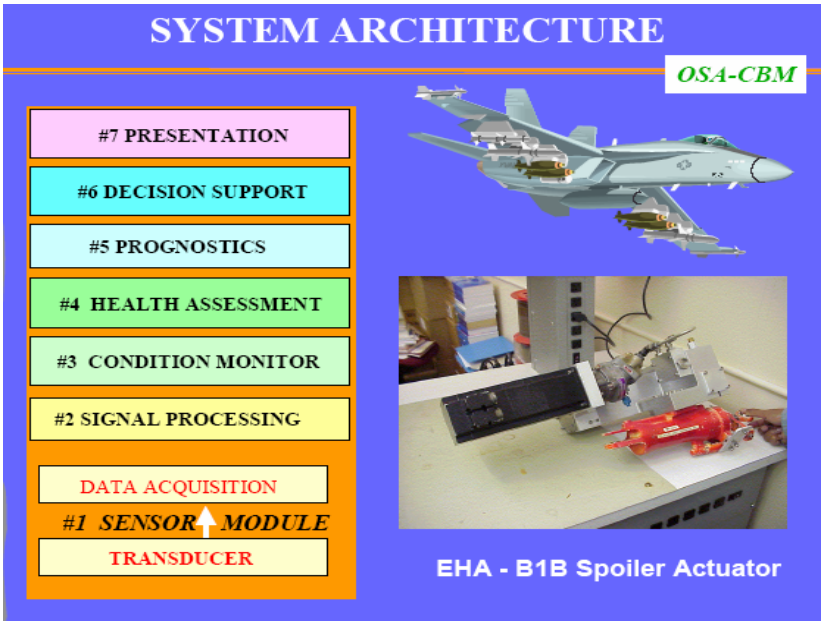


Figure 5.23: Health Monitoring of an Aircraft Actuator.

#1,2 Signal processing module generates digitally filtered sensor data, frequency spectra, virtual sensor signals and other features using the transducer data from the data acquisition module.

#3 Condition monitor generates alerts based on preset operational limits using the data from the sensor module, the data manipulation module and other condition monitors.

#4 Health assessment module determines the health of the monitored component/sub-system or system, generates diagnostic records and proposes fault possibilities. The diagnosis is based upon trends in the health history, operational status, loading and maintenance history.

#5 Prognostic module calculates the future health of the system based on the output of the health monitoring module.

#6,7 Decision support module generates recommended actions and alternatives.

The motor system setup and the typical faults considered for the health monitoring of the electro hydraulic actuator described above are shown in Figure 5.24 [McRuer, D., D. Graham, and E. Krendel]. The implementation uses FFT (Fast Fourier Transform) and neural nets for signal processing, limit analysis for condition monitoring and causal net for health assessment.

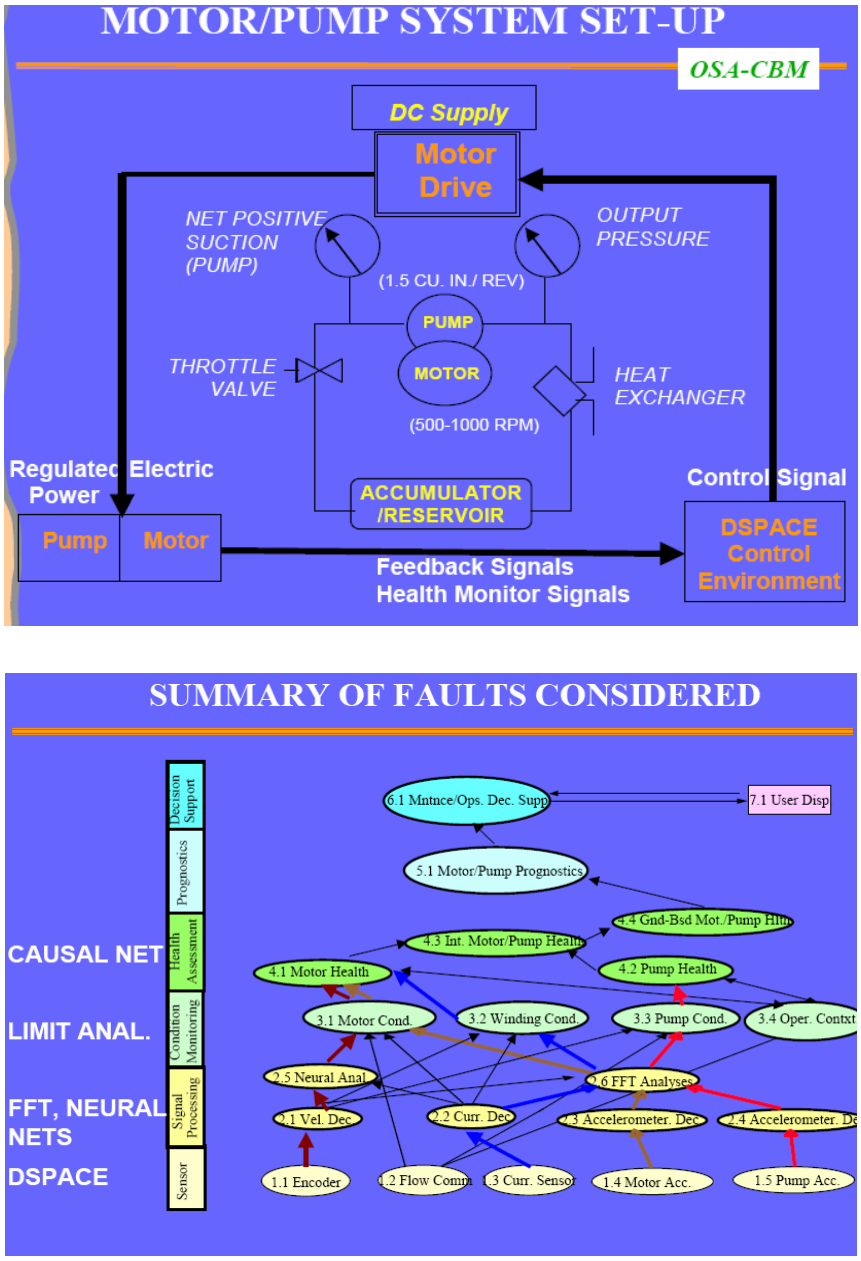


Figure 5.24: Health Monitoring System Example – Motor Pump Setup.

Several standards [McRuer, D. T. and E.S. Krendel] have been evolved for the development of systems based on OSA-CBM concepts.

5.9 INTEGRATED SYSTEMS HEALTH MANAGEMENT

Fully implemented, CBM+ will help predict a system's remaining operational life span, support operator decision-making, interface with control systems, aid in guiding maintenance repair actions, and provide feedback to the logistics support and system design communities, all of which are areas of concentration being impacted by AFRL's S&T research in Health Management. The AFRL systems engineering design concept emerging to satisfy CBM+ capability requirements is called Integrated Systems Health Management (ISHM). AFRL has embarked on a collaborative effort to develop a strategy and technology

development plan that will result in ISHM capabilities for Air Force (AF) systems. The AFRL ISHM initiative is planned to continue through fiscal year 2017 as depicted in Figure 5.25. ISHM encompasses the use of engineering, performance, test, inspection, and maintenance data, which is required to extract factors related to the various system and knowledge-based processes called out by CBM+. EHM is that sub-component of Integrated Systems Health Management that concentrates on the propulsion system.

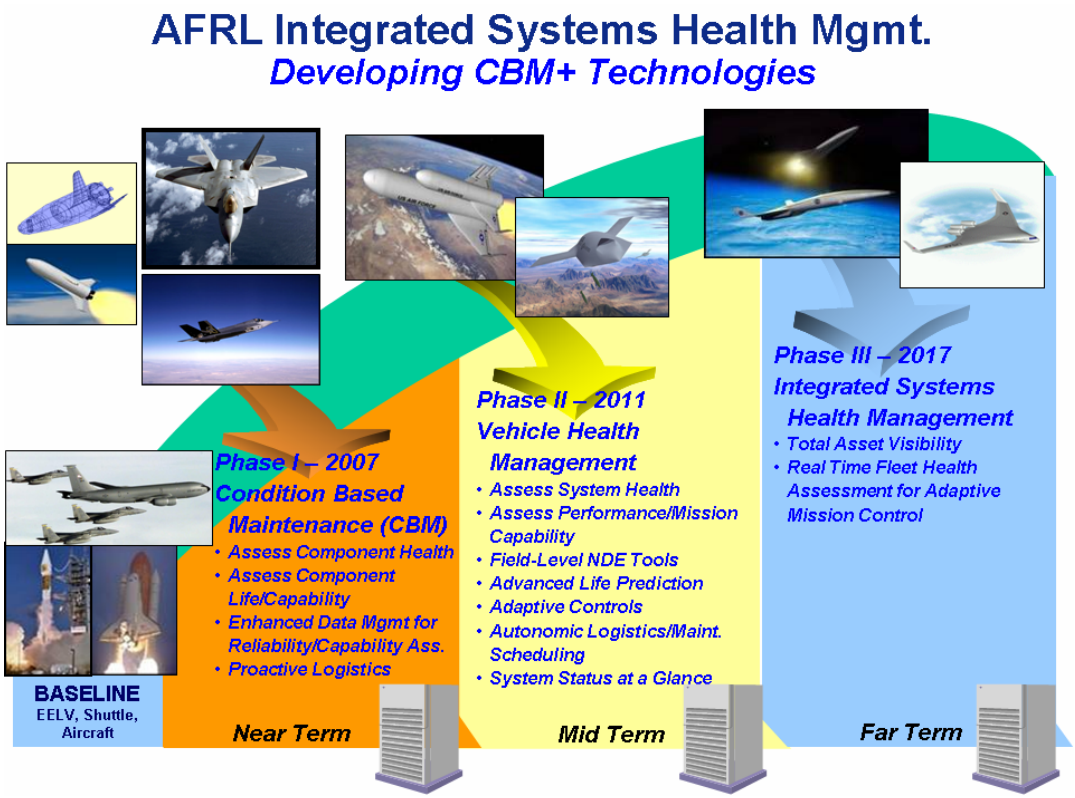


Figure 5.25: AFRL ISHM Program Initiative: Depicts AFRL ISHM CBM+ Technology Programs by Capability Development through Fiscal Year 2017.

Progression to ISHM under CBM+ will require changing business processes throughout the systems engineering process from the inception to the operation stage. To be clear, all disciplines involved, from S&T to development and sustainment, of the weapon system including support equipment and information systems, must forge new relationships and work together to realize the benefits of ISHM.

Under the ISHM umbrella, AFRL is seeking advanced capabilities such as enhanced prognostic and diagnostic techniques, advanced algorithms for failure trend analysis, electronic portable or point of maintenance aids, serial item management, automatic identification technology and data-driven interactive trouble-shooting and maintenance training. The ultimate ISHM vision is to have a fully integrated system operation, providing inputs and outputs for an autonomic system-of-systems, supporting both air and ground operations. The intent of ISHM is to increase operational readiness, and system performance, ensuring mission effectiveness regardless of the degradation state of the system. Payoffs include reduced life cycle costs by enabling a more responsive logistics system.

5.9.1 ISHM Research Requirements

The AF is transforming logistics processes to become lighter and leaner, to develop a more responsive planning and execution capability, to achieve an agile, effective sustainment process, and to develop

responsive, effective, fully integrated joint operations. Military success in the 21st century will require, advanced capabilities, superior speed, power, precision, endurance, and agility. The Science and Technology (S&T) community is working to develop technologies with substantial operational and agile combat support pay-offs. Laboratory technologies, being developed under the ISHM initiative, will revolutionize current AF maintenance philosophy by removing legislated intervals for programmed depot maintenance, by changing maintenance to only be when and where needed, and by implementing unified depot/field mission management concepts. The Laboratory tools and technologies will help Air Force logisticians obtain their desired goals for increasing equipment availability and reducing annual operations and support costs by 10%. The Laboratory is charged with helping the logistics community to reduce sustainment costs, so that more of the AF budget can be used for modernization.

AFRL is developing revolutionary data exploitation techniques, advanced prognostic and diagnostic capabilities, electronic portable maintenance aids, serial item management tools, automatic identification technologies, data-driven interactive trouble-shooting methods, new maintenance training concepts, adaptive planning modules, new sensors, and life prediction models.

In addition to linking AFRL ISHM program objectives to DoD and Air Force Strategic documents this research is conducted to benefit future weapon systems. Within the laboratory environment there are a plethora of programs targeting health management capability gaps that fall under the on-board and/or the off-board system management IT purview. These AFRL programs were born predominantly out of capability needs generated or expressed by weapon system acquisition programs for emerging system concepts, like the Long Range Strike Aircraft (LRS), or systems that are entering the early stages of systems engineering for development. These acquisition programs work with AFRL during their programs requirements development phase on establishing S&T requirements to address their known capability gaps.

A logistics concept being developed for sustaining future Air Force new weapon systems acquisitions is called autonomic logistics. The autonomic logistics concept entails automated automatic support capability needs via IT for sustaining support operations. In this regard, the IT system will be used by all support and operation functions required to manage equipment assets, create and implement mission plans, as well as maintain weapon- system- specific- pilot and -maintenance training records.

Additionally, this system as depicted in Figure 5.26 [McRuer, D. T. and E. S. Krendel] above will collect the gamut of data needed to support the data integration concepts required for ISHM and CBM+ to be fully realized. The prototypes and demonstrations resulting from AFRL ISHM programs will lay the foundation for system engineering of individual AFRL capabilities into the design of this future IT system.

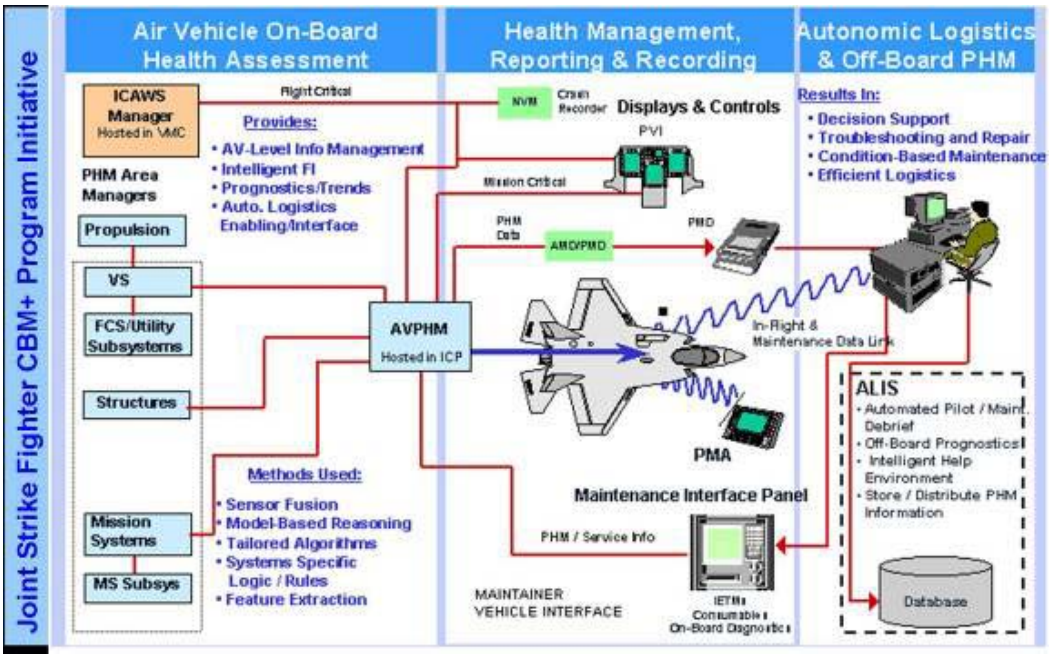


Figure 5.26: Vision for an Autonomic Information Infrastructure that Fully Integrates and Automates Logistics Functions.

The future weapon system programs discussed above are just examples of where AFRL managers obtain project ideas for research initiatives relating to ISHM IT needs. However, for most of these research programs, legacy system data is frequently used because acquisition weapon systems typically do not have or guard their data very closely or the data is not available for research purposes. Another data problem, in respect to new acquisition programs, is the lack of sufficient data needed to perform experiments that fully exercise newly developed AFRL e-tool capabilities. In fact, most of the AFRL IT [knowledge-based capability] research for new acquisition weapon systems is conducted using sample data from legacy weapon systems. Under this practice sufficient data can be obtained to test capabilities to the confidence level required for proving a concept and to satisfy the justification needed for a successful technology transition. Additionally, utilizing legacy systems data provides S&T programs a second target for potential capability transition.

5.9.2 AFRL ISHM Programs

Throughout AFRL, Directorates are creating and managing S&T programs for advanced on-board and or ground-based ISHM IT capabilities. These programs focus on developing advanced information fusion and data integration concepts, which use performance data from embedded sensors in combination with test, servicing, and repair data to determine vehicle or system health. Emerging from these concepts is the development of advanced algorithms to satisfy system and Information Technology e-tool capability needs. The resulting prototypes demonstrate innovative capabilities for needs such as trending and predictive failure analysis for maintenance servicing and repair recommendations.

The intention is for these AFRL ISHM developed capabilities to enable the maintenance community in moving forward to a proactive maintenance construct as called out in the new DoD 5000.2. It is envisioned that the health management concepts and/or products resulting from laboratory programs will enable maintainers to quickly assess and determine the best means to prevent or repair weapon system problems. The overarching ISHM ground-based system objective focuses on delivering capabilities for managers of weapon systems to ascertain and assess new information, through data exploitation, about the current condition of the equipment they manage fleet-wide. As AFRL health management concepts

mature, it is expected that new software application capabilities for Air Force logistics systems will also emerge.

The discussion thus far has covered very basic information on how and why AFRL program initiatives are generated and research targets are selected. There is much more planning, detail, and networking that go into those processes for program concept approval. Those process details will not be covered in this paper.

5.9.3 Obstacles to Health Management Research

Researchers require a significant amount of mature systems data to achieve a high level of confidence in the predictive methodologies, so that transition of a capability can occur. Yet, the primary obstacles to health management research are both data and technology transition related. For data, the issues are accessibility, data management policy, and the current data systems used in capturing data. The issues with data are so significant that AFRL managers begin coordinating and planning to meet data needs simultaneously to developing the research program content. The most pressing issue with transitioning advanced health management concepts to information technology programs is there lacks a formal process within the IT systems acquisition community for AFRL to engage in.

5.10 SYSTEM TEST, VERIFICATION AND VALIDATION AND CERTIFICATION ISSUES

Here we present validation, verification technique within the context of mission management.

5.10.1 Basic Definition

Verification: Ensuring that each step in the process yields the right product.

Validation: Ensuring the product will satisfy functional and other requirements.

5.10.2 Verification Techniques

There are many different verification techniques but they all basically fall into 2 major categories – dynamic testing and static testing.

- **Dynamic Testing** – Testing that involves the execution of a system or component. Basically, a number of test cases are chosen, where each test case consists of test data. These input test cases are used to determine output test results. Dynamic testing can be further divided into three categories – functional testing, structural testing, and random testing.
- **Functional Testing** – Testing that involves identifying and testing all the functions of the system as defined within the requirements. This form of testing is an example of black-box testing since it involves no knowledge of the implementation of the system.
- **Structural Testing** – Testing that has full knowledge of the implementation of the system and is an example of white-box testing. It uses the information from the internal structure of a system to devise tests to check the operation of individual components. Functional and structural testing both chooses test cases that investigate a particular characteristic of the system.
- **Random Testing** – Testing that freely chooses test cases among the set of all possible test cases. The use of randomly determined inputs can detect faults that go undetected by other systematic testing techniques. Exhaustive testing, where the input test cases consists of every possible set of input values, is a form of random testing. Although exhaustive testing performed at every stage in the life cycle results in a complete verification of the system, it is realistically impossible to accomplish.

- **Static Testing** – Testing that does not involve the operation of the system or component. Some of these techniques are performed manually while others are automated. Static testing can be further divided into 2 categories – techniques that analyze consistency and techniques that measure some program property.
- **Consistency Techniques** – Techniques that are used to insure program properties such as correct syntax, correct parameter matching between procedures, correct typing, and correct requirements and specifications translation.
- **Measurement Techniques** – Techniques that measure properties such as error proneness, understandability, and well-structuredness.

5.10.3 Validation Techniques

There are also numerous validation techniques, including formal methods, fault injection, and dependability analysis. Validation usually takes place at the end of the development cycle, and looks at the complete system as opposed to verification, which focuses on smaller sub-systems.

- **Formal Methods** – Formal methods is not only a verification technique but also a validation technique. A formal method means the use of mathematical and logical techniques to express, investigate, and analyze the specification, design, documentation, and behaviour of both hardware and software.
- **Fault Injection** – Fault injection is the intentional activation of faults by either hardware or software means to observe the system operation under fault conditions.
- **Hardware Fault Injection** – Can also be called physical fault injection because we are actually injecting faults into the physical hardware.
- **Software Fault Injection** – Errors are injected into the memory of the computer by software techniques. Software fault injection is basically a simulation of hardware fault injection.
- **Dependability Analysis** – Dependability analysis involves identifying hazards and then proposing methods that reduces the risk of the hazard occurring.
- **Hazard Analysis** – Involves using guidelines to identify hazards, their root causes, and possible countermeasures.
- **Risk Analysis** – Takes hazard analysis further by identifying the possible consequences of each hazard and their probability of occurring.

Scope:

For any system development several phases must be followed as shown in Figure 5.27. These phases are part of the verification, and validation process.

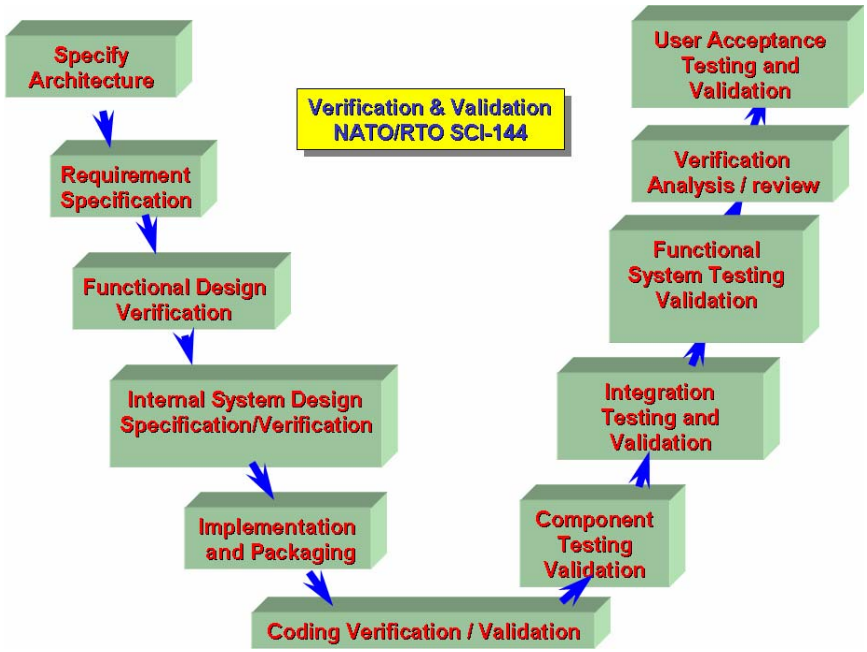


Figure 5.27: Generic Verification and Validation Process shown as "V".

The IEEE Standard for Software Verification and Validation (IEEE Std 1012-1998) contains information on software integrity levels, the V & V process, the Software V & V reporting, administrative, and documentation requirements, and an outline of the software verification and validation plan.

Verification and validation can be performed by the same organization performing the design, development, and implementation but sometimes it is performed by an independent testing agency. This is called independent verification and validation (IV & V). These agencies usually need to be accredited by a higher organization, to be sure that their results are dependable. For example, in the United Kingdom, the National Measurement Accreditation Service has begun to accredit companies for testing computer software used in safety-critical systems. The first company was accredited in 1994. The testing methods approved include a suite of in-house procedures including static and dynamic testing techniques.

5.10.4 Certification Process

Verification and validation are part of the long certification process for any embedded system. There are different reasons why a product needs certification. Sometimes certification is required for legal reasons. For example, before an aircraft is allowed to fly, it must obtain a license. Being certified would also be important for commercial reasons like having a sales advantage. One of the main reasons for certification is to show competence in specific areas. Certifications are usually carried out by independent agencies or other organizations with a national standing.

Certification can be applied to organizations or individuals, tools or methods, or systems or products. Certification of organizations aims at assuring that the organization achieves a certain level or proficiency and that they agree to certain standards or criterias. This however, is not applicable to all areas because while it is easy to measure the procedures of a company, it is much harder to measure the competence with which they are performed. So certification is usually applied to areas such as quality assurance and testing as opposed to design. Certification may also apply to individuals where workers must be certified in order to be a certain profession. This usually applies to workers such as doctors, lawyers, accountants, and civil engineers. Tools or methods may also be certified. For example, although DO-178B does not specifically define the tools that must be used, it does give certain requirements of tools used to gain certification.

Finally, systems or products may also be certified. [Storey96] In certification, there is always the issue of whether artifacts or methodology be certified. This becomes an issue in the certification of products containing software. Because software testing is so difficult, certification must be based on the process of development and on the demonstrated performance. This is a case where the methodology (development process) is certified instead of the artifact (software).

Review and analyses are performed on the following different components.

- **Requirements Analyses** – To detect and report requirements errors that may have surfaced during the software requirements and design process.
- **Software Architecture** – To detect and report errors that occurred during the development of the software architecture.
- **Source Code** – To detect and report errors that developed during source coding.
- **Outputs of the Integration Process** – To ensure that the result of the integration process is complete and correct.
- **Test Cases and their Procedures and Results** – To ensure that the testing is performed accurately and completely.

The 2 main objectives of the software testing process are to demonstrate that it satisfies all the requirements and to demonstrate that errors leading to unacceptable failure conditions are removed. The testing process includes the following three different types of testing.

- **Hardware/Software Integration Testing** – To verify that the software is operating correctly in the computer environment.
- **Software Integration Testing** – To verify the interrelationships between the software requirements and components and to verify the implementation of the requirements and components in the software architecture.
- **Low-level Testing** – To verify the implementation of software low-level requirements.

5.11 SUMMARY

This chapter describes the overall approach to develop technology for an intelligent mission management system to provide greater robustness. Robustness and fault tolerance are mission-driven requirements. As control technology progresses, improving system performance becomes more complex. Therefore, the price for achieving high performance and robust control is additional complexity of the controller. Fault diagnosis and fault tolerant control are critical component of complex systems.

Autonomous operation requires critical mission management capability for success. Consequently, data must be analyzed and interpreted quickly to yield results that can be implemented to enhance performance in mission operation. Automation can augment or replace human decision making in order to increase reaction speeds, reduce errors, stress, mitigate cognitive overload, enhance safety, and lower costs. While human decision making may not be totally eliminated, the human element fails to offer high capability under continuous stress and integration with logistics infrastructures. However, achieving the goals of automation will require verification and validation techniques to achieve success. Verification and validation is an essential component of a systematic approach.

We live in an information rich environment; however, success often depends on limited access to appropriate information. Information processing speed, accuracy and robustness are critical to achieving successful control and prognostic capability. Control is a critical technology for an integrated system management and is increasingly important in for intelligent operation and management. Control allows the

operation of autonomous and semiautonomous unmanned systems that keep the systems operational, as well as sophisticated command and control systems that enable robust, reconfigurable decision-making systems. The technology is still evolving and in the future we can expect to see the programming platforms for systems placing these techniques within the reach. Until then, it is reasonable to expect future distributed systems capability using electronic control. The increased processing power and intelligence is likely to improve and will lead to robust systems that will be more fault tolerant. The pervasiveness of electronics, communications, computing and sensing will enable many new applications for autonomous operation, but will also require substantial expansion of the current theory and tools.