Fault Management Design Strategies

John C. Day[i]
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr. Pasadena, California, 91109
and

Dr. Stephen B. Johnson [ii]
University of Colorado, Colorado Springs, Colorado Springs, Colorado 80918

Development of dependable systems relies on the ability of the system to determine and respond to off-nominal system behavior. Specification and development of these fault management capabilities must be done in a structured and principled manner to improve our understanding of these systems, and to make significant gains in dependability (safety, reliability and availability). Prior work has described a fundamental taxonomy and theory of System Health Management (SHM), and of its operational subset, Fault Management (FM). This conceptual foundation provides a basis to develop framework to design and implement FM design strategies that protect mission objectives and account for system design limitations. Selection of an SHM strategy has implications for the functions required to perform the strategy, and it places constraints on the set of possible design solutions. The framework developed in this paper provides a rigorous and principled approach to classifying SHM strategies, as well as methods for determination and implementation of SHM strategies. An illustrative example is used to describe the application of the framework and the resulting benefits to system and FM design and dependability.

I. Introduction

Development of dependable systems relies on the ability of the system to determine and respond to off-nominal system behavior. There are many factors that threaten the dependability of systems - systems are deployed in environments that are not entirely known in advance, complex systems have unintended emergent behaviors, and systems must continue to function, despite degradations, throughout their operating lifetime. The capabilities that provide dependability in a system are referred to as System Health Management (SHM). The operational (aka "run-time") aspect of SHM is defined as Fault Management (FM), which includes capabilities to contain, prevent, detect, diagnose, respond to, and recover from conditions that may interfere with nominal system operations. Or alternatively, the capabilities that address what to do when a system becomes, or is predicted to become unable to function as intended[1].

Specification and development of these FM capabilities must be done in a structured and principled manner to improve our understanding of these systems, and to make significant gains in dependability (safety, reliability and availability). As with systems engineering in general, where articulation of design goals and realizations is hampered by the use of natural language, dependability goals and the corresponding system capabilities must be formally stated and linked to the design[2].

A conceptual foundation for providing the needed mapping from dependability goals to SHM capabilities is provided in prior work[1]. This, in turn, builds on taxonomy and concepts developed in the dependability community over the past decades. Primary sources of dependability concepts and their relation to SHM concepts are found in earlier work by Avizienis *et al*[3] (basic concepts of dependability), Heimerdinger and Weinstock [4](application of

[i] Technical Group Supervisor, Systems Engineering Section, 4800 Oak Grove Drive M/S 301-490, AIAA Senior Member.

[ii] Associate Research Professor, Department of Mechanical and Aerospace Engineering, University of Colorado, Colorado Springs, not currently an AIAA member.

dependability concepts to generic systems) and Johnson and Day[5] (extension and relationship to SHM concepts). This conceptual foundation provides a basis for a framework to design and implement SHM capabilities that protect mission objectives and account for system design limitations.

Using this conceptual foundation, this paper develops a rigorous and principled approach to classifying SHM strategies, and a structured method for appropriate selection of a given SHM strategy for a given system objective. This method takes into account the real-world implications of re-use—systems are rarely developed entirely from a "clean sheet", and the use of patterns and elements from prior design has implications on the selection and application of SHM strategies. The classification of SHM strategies is supported by an illustrative example (using a familiar personal transportation analogy) to describe the application of the framework and the resulting benefits to system and FM design and dependability.

II. Goals and Function Preservation

System Health Management (SHM) is defined as "the capabilities of a system that preserve the systems's ability to function as intended."[6] In turn, functions exist to achieve system goals. Thus SHM capabilities exist to preserve the system's ability to achieve goals. The linkage between functions, goals, and SHM capabilities are clearly visible in a Goal-Function Tree (GFT) representation, as described in the 2013 Infotech@Aerospace paper, "Goal-Function Tree Modeling for Systems Engineering and Fault Management."[2]

The relationship of goals and functions is defined through the simple equation $\mathbf{y} = f(\mathbf{x})$, where the function f is a transformation or mapping of a set of input state variables $\mathbf{x}$ into the output state variables $\mathbf{y}$. A goal is represented by the system's output state variables $\mathbf{y}$ being constrained or controlled to be within some nominal range. The GFT, as its name implies, is a tree-like structure in the same sense as a classical functional decomposition or fault tree, with higher level goals and functions being decomposed into lower-level goals and functions. It represents goals and functions in a hierarchical tree structure. An important insight is that it is the intentions (goals) of the system's designers and users that provide a hierarchical priority structure to the system's intentions, and hence to its goals and functions. Essentially, engineered systems exist to perform achieve a particular goal or set of goals, and it is this goal set that comprise the top-level goals for the system. All other goals and functions exist to support the achievement of the top-level goals. It then becomes possible to assess and analyze various characteristics of the system with respect to this goal-function structure, in particular issues of completeness and coverage of system goals and functions.

For every system goal, there is the possibility that this goal cannot be achieved in the real system, due to a failure somewhere in the system. System Health Management addresses the measures taken to ensure the achievement of system goals. One way to address potential failure is simply to not allow failure to occur, which generally means beefing up the robustness and design margins in the relevant system components (i.e. higher-reliability components). The other way to address potential failure is to detect and respond to predicted or actual failure. These detection and response capabilities are new system capabilities, and hence are new system functions associated with new off-nominal goals. That is, detection and response capabilities are new off-nominal, SHM functions addressing off-nominal goals that can be represented in a GFT just as the nominal functions and goals are. These off-nominal goals and functions exist to protect and preserve the system's nominal functions. Just like the nominal active control loops, operational SHM functions are organized as active control loops, detecting off-nominal conditions, determining the cause(s) of these conditions, deciding which actions to take to mitigate these off-nominal conditions, and actions to preserve system function. The operational functions used in these control loops are presented and discussed in prior work,[1] and the set of functions used in a particular FM control loop depends on the nature of the goals to be protected. In this paper, we will often frame the discussion in terms of a structured, hierarchical, top-down design process. This process and its representation in terms of levels of the GFT decomposition provide a key basis and background for the description and selection of the SHM strategies that are the focus of this paper.

Another critical characteristic of the GFT is its reliance on state variables to provide physical and logical rigor to the representation of goals and functions. The association of state variables to all goals and functions ensures that the decomposition is physically and logically matched to the design, since design simply provides mechanisms to ensure that the functions are implemented and hence that goals are achieved. Put another way, the design provides the means by which the output state variables $\mathbf{y}$ are controlled within the desired range. Thus it is possible to create a comprehensive and physically accurate mapping from system functions and goals as represented in the GFT and the system design, which "realizes" or "actualizes" the functions in physical reality.

Finally, not all goals are equal in importance. Typically, it is possible to classify each goal (or objective) into one of the following three classes: Safety critical, mission critical, and non–mission critical. Safety critical objectives are those necessary to maintain the safety of humans. For some systems, safety also includes critical infrastructure. For systems such as space launch vehicles, this includes the launch pad infrastructure. Mission critical objectives are those necessary to accomplish the system's mission or highest-level goals (aside from human safety). For a space launch system, the mission is to deliver its payload (whether human or cargo) to orbit. For a deep space probe, the mission objective is to return science data to Earth. Finally, non-mission critical objectives are important objectives that are not necessary to achieve the mission, but are otherwise important. For a space launch vehicle, it is not necessary for the launch vehicle telemetry to be returned to Earth to achieve the current mission, but it is nonetheless important for the long-term cost-effective operation of the fleet of launch vehicles to relay information about the performance and health of system components. This information is used to ensure that the next flight is successful and more efficiently achieved. Each objective type has different levels of reliability and failure tolerance that are levied on that class. Safety goals are typically very stringent, while mission goals are somewhat less so, and non-mission critical goals least of all. These differing levels of rigor then influence the SHM strategies that are selected for their respective mission goals. The nature of a given objective constrains the possible set of protection strategies to be taken, and provides a means for determining the set of necessary (minimal) SHM functions to satisfy the program/project requirements.

III. System Health Management Strategies

Development of a set of defined SHM strategies allows systems engineers to manage the possible design options by focusing in the key concerns, and deriving design options from them. There is a small set of possible strategies to deal with possible failures in a system. We have found that the key differentiation between strategies is based on answering two primary questions with respect to protection of a system objective – "Will the design solution be focused on prevention of failure *causes*", and "Will the design solution seek to prevent failures, or tolerate them?" For most complex systems, the answer will include some of both.

The first question provides the basis for the first-order disctinction in differentiating SHM strategies. From the perspective of an operating system, focusing on failure *causes* implies a design-time solution, where potential failure causes are identified and mitigated as the design is developed and realized. The alternative is to address the effects of potential failures causes. We term the prevention of failure causes "Fault Avoidance", and the addressing of causes when they occur "Fault Management." These provide the labels for the categories of SHM strategies. Within the category of FM, we make a further distinction between actions that seek to prevent failures, and actions that mitigate failure once they have occurred. We define "Failure Avoidance" as the operational actions to prevent a failure from occurring (such as condition-based maintenance), and "Failure Tolerance" as the operational actions taken to mitigate the effects of a failure on system performance. These definitions and relationships result in the taxonomy shown in Figure 1. This taxonomy differs from the definitions articulated by Avizienis[3], but are conceptually related (our term fault avoidance is conceptually-similar to fault prevention, and fault management is similar to fault tolerance). We make these distinctions here as a way to differentiate in a consistent way between failure causes ("faults") and effects ("failures"), and the classification of strategies to apply to each. Avizienis does not distinguish, for example, between the strategies of failure avoidance and failure tolerance that we describe in this paper.

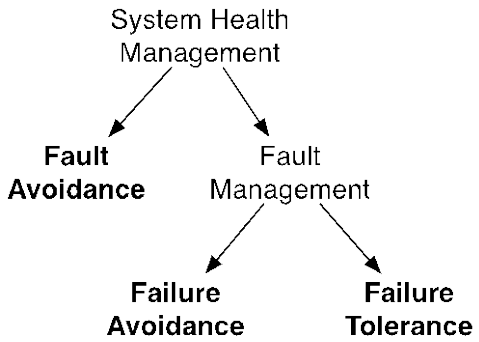

Figure 1: SHM Strategy Taxonomy. *Basic differentiation and terminology used to organize SHM strategies.*

We refer to a definition of *acceptable* performance that is tied to the definition of an objective (*aka* goal), which has both value and time elements. Acceptable performance is assessed by whether the performance of the function meets the set of constraints (as defined by the set of applicable goals) at a given point in time. A failure is the unacceptable performance of intended function, which implies a definition of acceptability[1]. The selection of SHM strategy depends on the determination of function criticality, as described in the prior section. In the process of

designing the system, there must be an assessment of whether a failure of the function is allowed (or, phrased less optimistically, impossible or too costly to prevent!). If a failure is allowed, a determination must be made as to whether the function must be recovered (within an acceptable timeframe for recovery, the time-to-criticality) or allowed to remain failed for some period of time. If a failure is not allowed (e.g., if the function is safety-critical and time-critical), then a strategy for preventing the failure from occurring must be determined.

A given system design has a set of interconnected components, each capable of performing one or more functions. These functions are used in concert to achieve the objectives for which the system was developed. These relationships are described graphically in Figure 2. This intent (the set of objectives) is expressed in a set of hierarchical goals (the goal-function tree). At a given level of abstraction, there are a set of actions that satisfy the set of goals – these actions can be referred-to as a "plan." A given plan is one possible path to satisfying the existing set of goals, and there may be many plans that satisfy the goals, with typically a single plan being identified as the "nominal" plan[7]. These actions form the basis for lower-level goals in the planning decomposition, and inform the lower-level goals and functions in the GFT. To address a current or incipient failure in a system, there are generally only a small number of potential mechanisms available – (1) change the physics of the system (by removing the failure cause, inhibiting failure effect propagation, or engaging compensating mechanisms), (2) change the current plan or (3) change the set of goals. Note that a change to a plan can be distinguished from changes to the higher-level goals it is intended to satisfy, but because the selection of a specific plan results in the elaboration of lower-level goals, these goals will change when an alternate plan is selected. Put another way, when assessing any specific goal within the GFT, with respect to that goal all lower level goals and functions beneath that goal can be considered "the plan" to meet the goal. Thus what the designer or analyst or operator considers as "the goal" and "the plan to meet that goal" is merely a matter of personal or organizational choice. The terms can apply to any goal in the GFT. The application of these possibilities before or after the occurrence of a failure determines, in part, the label (either a goal or part of the plan to meet a goal) associated with the strategy and actions.

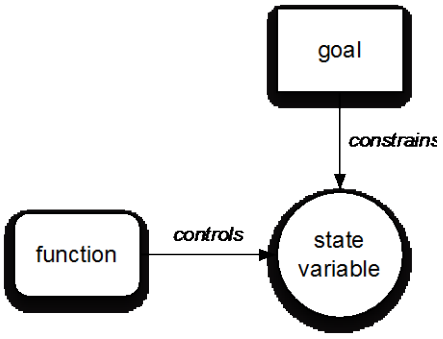

Figure 2: Relationship of Goals, Functions and State Variables

Given the conceptual organization presented in Figure 1, and the set of potential actions, a set of SHM strategies has been developed, and each is described in more detail in the following subsections. Each strategy implies a different application of SHM functions in the system, and has implications on the allocation of SHM functions in the system design. Multiple strategies may be applied in the preservation of a given function or goal. As a way of illustrating the concepts, a simple example using an automobile is described with each SHM strategy.

Fault Avoidance

Fault avoidance is the SHM strategy that prevents the causes of failure from occurring. For causes internal to the system, which by definition are the only ones that can be controlled by the designer or operator, failure causes are "faults." Typically, the fault avoidance strategy relies on design-time processes such as quality assurance and/or design features such as large margins. Fault avoidance is based on the idea that the application of processes during design-time can detect and remove flaws in components, designs and assumptions, thereby eliminating or reducing the probability of possible failure causes. There are no operational FM functions applied. Used in the example of a car, we would label the inspection of build quality on assembly line or incorporation of design changes in next year's model as fault avoidance strategies.

While design-time quality measures are typically used for fault avoidance, there is also an inherent assumption that the system will not be operated out of its specified design limitations. For the typical car, this means that the driver will not exceed certain speed limits, will not ram the car into fixed structures, will not drive it over a cliff, etc. Thus the fault avoidance strategy implies that the system operators will "preclude" certain classes of operational failures. Car designs do assume and account for the possibility of accidents and attempt to keep the driver safe (this is a second goal; cars are intended to transport passengers from one location to another, but a second goal is to ensure those passengers are "functioning properly" at all times), but no design can mitigate operator-created failures beyond a certain level of severity. These must be prevented.

Failure Preclusion

Failure preclusion is the SHM strategy that results in the application of operational FM functions to preclude a failure from occurring. Failures are predicted based on the observation of current system performance (including any observed degradation), interaction with the environment, intended system actions, and the propagation of present and expected conditions into the future. Failures are avoided by using the prediction to determine and take action that precludes failure of critical functions before they occur. Condition-based maintenance and alteration of a spacecraft pointing profile to avoid a potential component over-temperature are examples of failure prediction and preclusion. An associated analogy for a car is the periodic observation of tire wear , such as measuring remaining tread and assess upcoming driving conditions to determine when to change tires before they become unsafe.

It should be noted that it is not always obvious when an operational measure is just "normal operation" of the system, and when it is an FM strategy. For example, if a car is not periodically refilled with gasoline, it will run out of fuel and stop operating; that is, it will fail (to function). Most people would say that refueling a car is part of the normal "plan" for operating a car, and is not considered "maintenance". However, what is to distinguish this from the "maintenance operation" (an FM function/strategy) of replacing a tire or a fan belt, since these are also "in the plan" of car operation? There may not be a useful distinction in classifying these actions separately—there are differences in frequency and observability in the example above, but in either case action must be taken to avoid failure. That is, part of failure preclusion is executing the plan! This illustrates the idea that FM is an integral part of systems engineering, and that sometimes somewhat artificial distinctions are made between "nominal" and "off-nominal" system functions.[iii]

Failure Recovery

Failure recovery is the SHM strategy that allows a failure in the system to occur and to temporarily impinge on the system function to which the strategy is applied, with the intent of correcting the failure to enable continuation of the the existing or redefined mission. In this approach, a set of FM operational functions are used to detect the failure and restore acceptable function performance. The failure of the function is the triggering event, and the FM response attempts to restore that function by returning the affected states to acceptable performance values. There is a spectrum of possible outcomes as a consequence of failure recovery actions. At one end, the recovery time is insignificant and the overall goal can still be achieved completely, even though the system's performance is "unacceptable" for some period of time while the recovery actions are performed and take effect. At the other end of the spectrum, the failure recovery takes so long that all or part of the original goal(s) are no longer attainable, and either the compromised partial goal is accepted, or an entirely new goal can be put in its place.

Take the example of a driver who intends to reach a city 120 miles away, but a tire unexpectedly fails halfway to his destination. The driver pulls over (a goal change/safing strategy—see below), replaces the tire with his spare (the first part of the failure recovery), re-starts the car and accelerates and gets back on the road (the second part of the failure recovery) and proceeds to complete the trip (which is an activity that was part of the original plan). The function has been restored, and the driver reaches the objective, although potentially delayed. If the delay time was such that the goal of the trip was fully achieved, then no overall goal change occurs. However, if the delay is such that the driver is late to an important meeting, then there is a compromise of the original goal to attend the entire meeting. If the delay was such that the entire meeting is missed and the driver returns home instead, then the mission is "aborted" altogether in favor of a new goal to return the driver safely home. In this case, the failure recovery is a necessary step to enable completion of the redefined goal.

Failure Masking

Failure masking is the SHM strategy that allows a failure in the system to occur, but through application of FM functions, the failure effects do not propagate to impinge on the critical system function to which the masking strategy is applied. This is usually performed by detection of a low-level failure, and containment of the failure effects through actions or functions that contain the propagation of the failure effects and compensate for the failure. These actions either change the physics of the system to alter the propagation of failure effects, or have sufficient margin to absorb the failure effects. Typical aerospace examples are applications of *n-of-m* redundancy schemes such as voting logic, extra capacity in batteries and solar cells and use of 4 reaction wheels for 3-axis control. Typical cars do not use failure masking, but there are examples of large trucks that have multiple tires, that allow the driver keep traveling with a failed tire (albeit with less load capacity), and future cars may have redundant computers to control time and safety-critical functions such as braking, if these become fully computer-controlled.

[iii] We thank Dr. Eric Barszcz from NASA Ames Research Center for first describing and highlighting the subtleties involved in this example.

Goal Change

Goal change is the SHM strategy that modifies the set of system goals in an attempt to define goals that are achievable given the observed and predicted condition of the system and environment. This is applicable both in a reactive sense (responding to a detected failure) and a predictive sense (responding to an impending failure). This strategy presumes that the full set of current or future goals are not achievable under the current failure condition, or that there is sufficient uncertainty in predicted performance to change the intent. Typically the modified goals (of which there may be few or many) are less demanding than the original set (e.g., transition to a Safe mode), or are mechanisms to protect some goals that are more important than others (e.g., a launch vehicle ascent abort preserves crew safety, but abandons the goal of placing the crew in orbit). Using the car analogy, a goal change occurs when a driver, who intends to go to the grocery store, suffers an engine failure, and has to call a tow truck to take the car to a mechanic instead. The original intent is abandoned (or deferred to the future where the car is functioning again), and a new goal—have engine fixed—is put in its place.

Often, a goal change is applied as a near-term strategy so as to enable system repair, at which point all of the original goals could be restored. In the car example, the goal change takes the car to the repair station instead of the original destination. Then once the car is repaired, it is once again able to achieve all of its goals. In the case described above in the Failure Recovery section, pulling the car over is a safing action to protect the driver as well as the car; it is also a necessary step to enable replacement of the flat tire. Similar examples exist for spacecraft, which frequently go into safing modes to protect the spacecraft and contact Earth so that mission operators can repair the spacecraft and continue the mission. The resulting mission might maintain all of the original objectives, in which case the repair can be classified as a full failure recovery. Or the mission might continue only to a smaller or different set of objectives if the original objectives can no longer be met. In this case, there is a global goal change for the mission, and the failure recovery is to the new goal.

IV. Strategy Nuances

While SHM and FM strategies appear straight-forward on first appearance, to apply them in practice, one must be aware of certain nuances and complexities. These nuances are typical for SHM/FM in general, coming into play whenever it is necessary to understand existing designs, or in creating new ones.

A. Viewpoint Relativity

One common issue is the problem of relativity of viewpoints, particularly with respect to different levels within an organization or project. One of the tenets of systems theory is that the theory can be profitably employed at many different levels of a system, whether at the total system level, the "segment" level (such as the vehicle versus ground segments, as described in military parlance), at the level of elements (such as different rocket stages), subsystems, components, etc. Because these hierarchical levels often correspond to organizational and institutional levels (organizations are often assigned to build or integrate a specific system, segment, element, subsystem, component, etc.), these different institutions necessarily utilize a view local to their specific task. For an organization building a subsystem, their subsystem "is" the system, but from a higher-level point of view, it is only a portion of the system.

This sort of issue occurs when considering the FM strategies being used within an existing system, or when determining what strategies to use for a new system. Two examples help to illustrate.

Consider first a launch vehicle with multiple liquid propellant rocket engines (LPREs), which have sufficient performance to enable an "engine out" capability for all or some of its ascent through the atmosphere. Engine out means that one of the engines can be shut down, but the remaining engines can be used (throttled up, for example, or simply utilized in a different control algorithm that recognizes one of the multiple engines is not functioning) in such a way so as to enable the launcher to deliver its payload to the nominal orbit (though most likely at a later time and at a somewhat different position than the nominal ascent). Assume that the LPREs have the capability to respond to incipient catastrophic failures by detecting the problem before the failure "goes uncontained" (that is, it blows up in a conflagration, fireball, or explosion), and shutting the engine down before a catastrophic failure occurs. Seen locally at the level of the LPRE, this is a goal change: the engine no longer performs its original function because it has been turned off. However, from a higher-level point of view, the mission might yet succeed in delivering the payload to the correct orbit. If this is the case, then we either have a failure and failure recovery response—this would be true if the monitored state variables, such as vehicle attitude errors or attitude rate errors, are temporarily compromised, or we have failure masking, if the engine problem shows no symptoms at the level of vehicle attitude

errors or attitude rate errors. Thus a goal change at the local level of the engine is seen as a failure masking or failure recovery at higher levels. This is often the case when considering typical "redundancy management," or "failure detection, isolation, and recovery/response (FDIR)."

Another typical situation shows a different logic, where the system level response is a goal change, followed in time by a failure recovery. Consider an Earth orbiting satellite or deep space spacecraft, in which a failure occurs that requires the spacecraft to engage a "safing" action. Safing usually means that the spacecraft shuts down all equipment not necessary to basic spacecraft survival, and then puts itself into a state in which it can send and receive data from mission operators on Earth to determine what should be done next. If the spacecraft was originally performing its nominal science mission when the failure occurred, such as performing stellar or planetary observations, then switching to a safe state is a goal change. The vehicle's goal has been changed from performing its science observations to protecting the vehicle and ensuring communications to Earth.

However, this cannot be the final action from the point of view of the entire system. While the vehicle segment has put itself into a safe state, this generates new activity in the ground segment, to diagnose the problem and then ultimately to initiate a failure recovery action. Thus the system as a whole is engaged in FDIR, and the ground segment (mission operators) will send up a sequence of new commands to re-start the science mission. If the regular mission can continue, then the failure recovery strategy succeeds fully in recovering to the nominal mission (no system objectives have been compromised). If the original failure has permanently degraded the capability to achieve all mission goals, such as permanent failure of a science instrument, or a loss of data during a one-time planetary flyby, then the new nominal mission can only achieve a lesser set of goals; the final degraded mission is a goal change from the original mission objectives. In this case, the difference in perspective is both hierarchical, in the sense of vehicle versus system perspectives, and temporal, in that in initial strategy of safing (goal change) is only part of a longer time-scale failure recovery strategy.

Temporality and hierarchy are also key factors in differentiating between failure masking and operational failure avoidance strategies. A prototypical failure masking response is the use of a computer voting scheme to mask the failure of one computer out of a suite of three or four identical computers for real time control. As noted above, classical FDIR can also be seen as failure masking, from a high-enough point of view (that is, seen from the point of view of a state variable that is "high enough" in the hierarchy of system goals, the state variable is not affected by the lower level failure). If a state variable has not been compromised, one might also say that a failure has been avoided in system operation.

This is the sort of situation typically associated with prognosis of a future failure, and repair or replacement of a component before it fails—that is, operational failure avoidance such as replacing a car tire before it wears so thin that it fails. We can also consider failure masking to be essentially vehicle-based operational failure avoidance. In many cases of vehicle response, there is an element of prognosis occurring, in which one aims to detect precursors of failures that could compromise the entire vehicle, and respond before a vehicle-level goal is compromised. Detecting a precursor necessarily means that one predicts that if the observed precursor failure is left unmitigated, then it will in the (near) future lead to system failure. We therefore want to detect and respond before this vehicle-level failure can occur. If to do this one uses a cold backup to be turned on and then replace a failed component, there is little to distinguish this situation from "normal" prognostics in which ground-based assets and teams predict that a failure will occur in the future, and ensure condition-based or schedule-based maintenance to replace the component before it fails. From a temporal perspective, the failure masking and prognosis/operational failure avoidance look very similar. However, one could potentially distinguish between a vehicle-based response and a system response requiring both vehicle and ground-based capabilities, or in the case of schedule-based maintenance, a purely ground-based capability.

B. Operational versus Design- and Manufacturing Time Viewpoints

Another distinct variant of the issue of viewpoint relativity is the distinction between how SHM strategies appear during operational use of the system, versus during design time and manufacturing phases of the system. The primary example of this is the selection of the "Fault Prevention" strategy. What this means is that during system operations, the causes of failure (faults) are, for a given goal and/or function, to be prevented from happening at all. Typically this means that the system is designed and built to have significantly increased design and environmental margins. Examples of this include:

1. Extremely thick, heat-resistant, radiation-resistant walls for nuclear reactors,
2. A spacecraft flying to Mercury or Venus with extremely white paint to reflect solar heat,
3. Electrical wire solders with very low probability of debonding,
4. Requiring that an automobile be operated in conservative ways that do not overheat the engines or cause loss of vehicle control.

The first three examples do not require any operational measures, whereas the operational constraints on automobile obviously require that the driver limits his or her performance demands on the system. In the first three cases, the operational view of the system is that these goals and corresponding functions are purely passive. However, there was significant activity in design or manufacturing to ensure that the increased margins and decreased probability of failure are achieved. In the case of the nuclear reactor wall, the designers took special care to address issues of structural thickness, heat, and radiation in reactor failure cases into account when selecting wall materials and determing wall thickness. In the Mercury/Venus spacecraft case, issues of heat reflectivity must be accounted for, and the painting of the spacraft itself must be performed in such a way that no unpainted areas remain. In the case of electrical wire solders, the primary measures include proper training of wire solderers (or programming of the relevant machinery) and inspections and tests of the resulting solders to ensure proper bonding. In the latter two examples, there are specific "closed loop measures" taken during design and manufacturing to double-check the manufactured items to ensure proper implementation of the design. In all cases, there are generally double-checks to be sure that the design addresses the proper set of requirements to account for the various issues and constraints that the design must address. In case number 4, there are not generally any design operational constraints placed on the vehicle operators (other than social/legal limits!), but in other cases such as mission operations of a spacecraft or pilots of commercial airliners, there could be procedural or design constraints put into the system to ensure operational compliance.

Note that for examples 2 and 3, that the achievement of a passive operational goal and corresponding passive functions requires active closed-loop control measures to be implemented in manufacturing. In examples 1, 2, and 3, there is the need for active closed-loop control measures to ensure compliance of the design with the requirements placed on the design. Thus fault avoidance in operations requires failure (detection and) recovery during design and in manufacturing. A designer or manufacturer must focus on the closed-loop activities that they must do to ensure detection and recovery from failures they might create in the designed or manufactured item, so as to enable the passive fault avoidance strategy for operations.

The common theme of the subtleties described in this section is that some of the issues that arise when trying to determine what SHM strategy to apply are due to the point of view of the one who is determining or labeling strategies. It is crucial to recognize the existence of these points of view to avoid confusion.

V. Strategy Development

A system's set of SHM capabilities exist to preserve the system's ability to achieve the designed intent. This intent can be specified in goals as documented in a GFT. Goals can be categorized according to criticality, and this assessment is used as part of the process of determining the SHM strategy (or strategies) used in the protection of the goal. Except for fault avoidance, each identified strategy implies a set of functions to be applied in the realization of the strategy – we label these operational SHM strategies as "FM strategies."

Each FM strategy ought to improve the reliability/availability/safety of the system, and hence the probability of success of achieving the associated goal. Multiple FM strategies are typically applied to protect a given goal, due to effectiveness and coverage limitations of each strategy. A given strategy may only apply to a subset of failure causes, requiring multiple strategies to achieve the necessary reliability. Multiple strategies can also be applied in a tiered sense, to designate a series of response action that have a preferred order (e.g., issuing a device reset [recovery] as a first step before swapping hardware or aborting the mission [goal change]).

Since the available design space for a given system is not a "clean sheet," new design solutions always use patterns and approaches from prior engineering designs (and in many cases, re-use of hardware and software components). When system elements re-use earlier designs, with those designs come previously-defined SHM strategies. This implies that the options in the selection of FM functions are limited by these historical constraints, and entire sections of the GFT are therefore predefined as well.

The selection of FM strategies is done on a per-goal basis. This is because the goals capture the intent and needed performance level for the current plan, and this is the appropriate basis for determining whether a response action should be taken, and of what type. It is not correct to define the FM strategies to apply at a given time based only on functions. For example, if functions alone were used as the basis for SHM strategy selection, the specified FM response action would be implied whenever the function failed, regardless of whether the function was needed to achieve the present system goal set (the failure of a function is only relevant when an active goal is assigned to the state variables being controlled by the function). In addition, the set of goals that define the required output of a given function define acceptable performance of the function—this provides the basis for classifying performance as nominal or off-nominal, and therefore the need for additional action. Furthermore, depending on the classification of

the goal in terms of its importance to safety and mission success, a function may require one strategy (e.g., failure avoidance) in one mission phase, while a different strategy (e.g., failure tolerance) would be more appropriate in another.

A decision must be made for each goal regarding which SHM strategy (or strategies) ought to be applied. This is a necessary step in circumscribing the trade space of system functions and components necessary to realize the strategies. The following steps outline the elements of a decision process for selecting a minimal SHM strategy, based on general criteria that limit the available strategy options. This does not exclude the possibility of specifying strategies beyond the minimal set. These steps are also shown graphically in Figure 3.

1) **Assess goal achievement requirements.** Perform analysis of current design to determine the current probability of meeting the goal (reliability), the anticipated time to recover (availability) and any requirements on failure cause tolerance (single-fault tolerant, etc.). If the current design is sufficient with respect to these requirements, then no design changes need to be applied. However, if they are not met, then additional changes to the design must be considered, which may include definition of additional FM strategies.
2) **Determine goal criticality.** The three classes of goals defined in Section II provide a basis for limiting the appropriate FM strategies to select for a given goal. If a goal is designated non-mission-critical, then the minimal FM strategy is goal change. The implicit assumption in this decision is that achievement of the remaining mission-critical and safety critical goals is improved by removing or changing the non-mission-critical goal.
3) **Determine time margin available.** The time margin available is based on the race condition between the failure effect propagation time and the predict/detect/react time of the FM control loop. If insufficient time margin is available, then failure masking is the only available option. However, time margin can be gained by either moving the detection upstream, or by moving the goal out in time (if this is an available option). Creating more time margin allows for other SHM strategies to be considered.
4) **Determine allowable period of outage.** Based on the physics of the current system design (which could be performed using a state effects diagram), determine the allowable period of outage for a given goal. For safety-critical and most mission-critical goals, this time is often zero (no allowable period of outage). For these cases, the choices for an appropriate FM strategy are limited to failure preclusion or fault avoidance. For other mission-critical goals and for non-critical goals, a non-zero recovery time opens up the possibility of applying a failure recovery strategy, which may offer opportunities for reducing the cost or complexity of the system design.
5) **Assess ability to predict.** If the degradation of function is both slow enough for operational adjustment, and these effects are observable within the system, then failure preclusion can be applied to preserve the current set of goals. This involves the projection of future state based on the estimated current state, the physics of the system and any planned changes to system configuration. If the function degradation occurs too quickly for an operational response, then the only remaining option is to rely on the non-operational processes that provide fault avoidance. Design changes to the system may allow improvement of both the observability and the time margin (e.g., Addition of sensors to increase observability), but these may incur increases to cost or complexity of the system. In the case where the observability and/or time margin are insufficient or too costly to implement, achievement of the required performance must be achieved through fault avoidance.

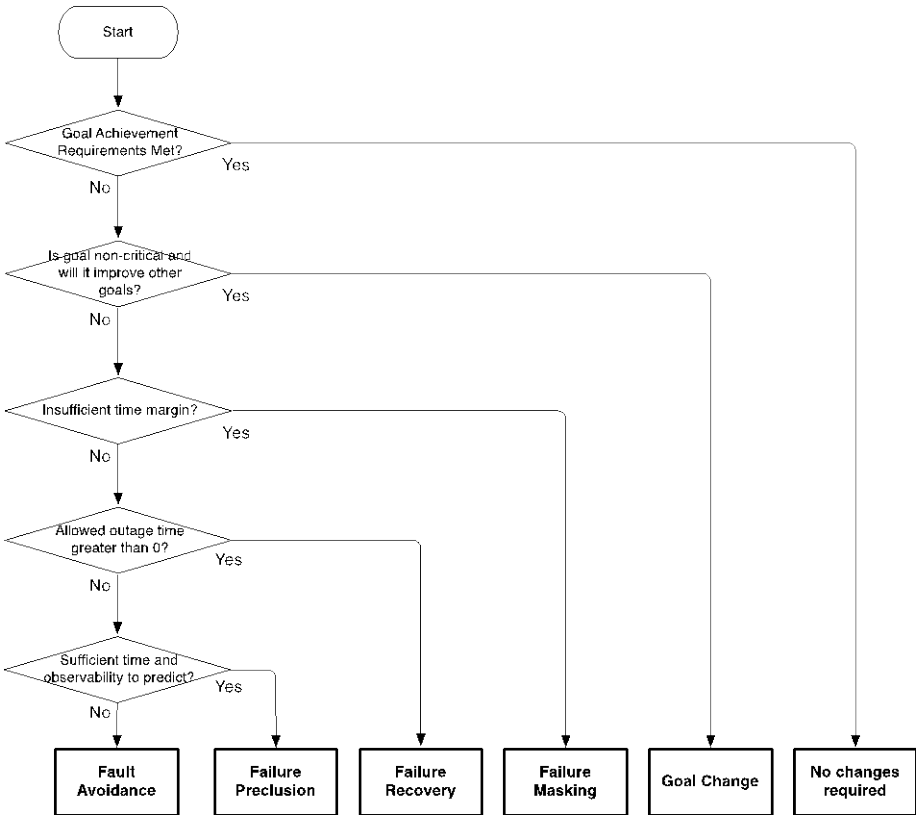

Figure 3. Criteria for Determination of Minimal SHM strategy.

This process allows the delineation of a minimally-acceptable selection of FM strategy for each goal, and is based on the relevant requirements associated with that objective (function preservation approach, required reliability and availability, ability to add types/levels of redundancy, etc.). Note that to meet FM performance requirements, or to work within a given set of design constraints, that multiple SHM strategies may be applied to each goal.

Realization of FM strategies also involves trades that depend on the available design space, and the cost of the implementation (mass, power, operations cost, maintenance cost, etc.) All operational SHM strategies require some form of redundancy to implement—masking requires redundant sources of information, goal change requires temporal redundancy, etc.) Further, if the failure avoidance strategy does not involve masking of failure causes, then some model of failure prediction must be employed. Ideally, these models are made explicit, but many times they are not. Schedule-based maintenance, as a failure avoidance strategy, generally results in a less-complex operational system, but is not robust to changes in the assumptions used to develop the maintenance schedule, or changes in the environment. Condition-based maintenance can address this concern, but it requires observability into relevant states of the system and operations-time calculation of remaining life. Both require redundancy in the form of spare units that can be used as replacements. On-board changes to the plan (such as: "use IMU2" instead of "use IMU1" for the rate sensing) are mechanisms to perform failure avoidance or recovery, depending on when they are applied.

VI. Conclusion

As the means of ensuring that system goals are achieved in the face of failure, SHM is an integral part of systems engineering. As such, SHM must be integrated into the systems engineering process. Development of dependable systems relies on the ability of the system to determine and respond to off-nominal system behavior. This paper describes the strategies (and a strategy classification structure) that systems engineers can apply to protect system goals and functions in a hierarchical structure of system goals as defined in a GFT. The strategy classification is

supported by an illustrative example to describe the application of the framework and the resulting benefits to system dependability. These strategies can be viewed from differing perspectives within the system, and the importance of taking into account the use of patterns and elements from prior designs is noted. Once these differences of perspectives are understood, systems engineers and SHM designers can apply the criteria defined in Section V of this paper to determine which sets of strategies can be used to protect system goals and functions at each level of the system goal-function hierarchy, starting from the system's top goals and functions, and working down through the GFT. This process provides a basis for defining SHM capabilities that improves the dependability of the system while accounting for system design limitations.

Acknowledgments

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Johnson, Stephen B., and John C. Day, 2011. System Health Management Theory and Design Strategies, AIAA Infotech Conference, St. Louis, Missouri, April 2011; AIAA paper 977233.

[2] Johnson, S. B., 2013. Goal-Function Tree Modeling for Systems Engineering and Fault Management, AIAA Infotech@Aerospace Conference, Boston, MA, September 2013.

[3] Avizienis, A., Laprie J.-C., Randell, B., and Landwehr, C., 2004. Basic Concepts and Taxomony of Dependable and Secure Computing, IEEE Transactions on Dependable And Secure Computing, 1 (1), 11-33.

[4] Heimerdinger, W. L. & Weinstock, C. B., A Conceptual Framework for System Fault Tolerance (CMU/SEI-92-TR-33, ADA264375). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992.

[5] Johnson, Stephen B., and John C. Day, 2010. Conceptual Framework for a Fault Management Design Methodology, AIAA Infotech Conference, Atlanta, Georgia, April 2010; AIAA paper 227006.

[6] Johnson, S. B., et al., eds., *System Health Management: with Aerospace Applications*, John Wiley & Sons, Chichester, UK, 2011, p. 3.

[7] Lamsweerde, A., *Requirements Engineering*, John Wiley and Sons, Chichester, UK, 2009, Chaps. 7, 9, 13, especially section 7.6.1.