Autonomous Prognostics and Health Management (APHM)

Jacek Stecki[1], Joshua Cross[2], Chris Stecki[3] and Andrew Lucas[4]

[1,3]PHM Technology Pty Ltd, 9/16 Queens Pde, VIC 3068, Australia.
jstecki@phmtechnology.com
cstecki@phmtechnology.com

[2,4]Agent Oriented Software Pty. Ltd., 580 Elizabeth Street, Melbourne, VIC 3000, Australia.
Joshua.cross@aosgrp.com
Andrew.Lucas@aosgrp.com

ABSTRACT

The objective of this paper is to show how PHM concepts can be included in the design of an autonomous Unmanned Air Vehicle (UAV) and in doing so, provide effective diagnostic/prognostic capabilities during system operation. The authors propose a PHM Cycle that is divided into two parts, covering the design of the Autonomous PHM system and the operation of the PHM system in real-time application. The paper presents steps in design of Autonomous Prognostics and Health Management (APHM) developed using the above approach, to provide contingency management integrated with autonomous decision-making for power management on a UAV. APHM was developed using commercial software tools such as the JACK® autonomous software platform to provide real-time intelligent decision making and MADe® - Maintenance Aware Design environment to identify risks due to equipment failures and to select appropriate sensor coverage. The PHM Cycle methodology is demonstrated in application to autonomous, real-time, engine health and power management on an Unmanned Air Vehicle (UAV).

1. INTRODUCTION

Prognostics and Health Management (PHM) is a new approach to enhancing system sustainability which redefines and extends Condition based Maintenance (CBM) on the basis of current advances in failure analysis, sensor technology and AI based Prognostics (Scheuren, W. J., Caldwell, K. A., Goodman, G. A. & Wegman, A. K. (1998)). The two basic tenets of Prognostics and Health Management are:

- Prognostics - predictive diagnostics which includes determining the remaining life or time span for the proper operation of a component
- Health Management - the capability to make appropriate decisions about maintenance actions based on diagnostics/prognostics information, available resources and operational demand.

The paper discusses the methodology for integrating PHM concepts into system design to provide autonomous diagnostic and prognostic capabilities during system operation

The Autonomous PHM system proposed in this paper is designed on the basis of correct risk assessment and the reasoning capability which is able assess the sensor readings and determine the state of the system and the appropriate action. The effectiveness of a PHM system depends on comprehensive and correct identification of risks due to system failures and system responses to those failures. Knowing the failures. the optimum combination of sensors must be identified and any ambiguities in the detection of failure modes resolved. Sensor coverage can be augmented by BITs and component-specific sensors to increase reliability of diagnostics and to eliminate ambiguities in the detection of failure modes. The resulting sensor set provides sensing patterns which are syndromes of particular failures of the system, and can be expressed as diagnostic rules. Diagnostic coverage maybe further enhanced by application of probabilistic methods. Having identified the functional failure modes and determined their criticality, reasoning techniques based on artificial agent technology can be applied to determine a set of actions that is the most appropriate for the given situation.

A reasoning system improves the diagnosis by maximizing the likelihood of determining the failure mode correctly. It is also able to determine the most appropriate course of corrective actions – taking into account current circumstances such as the flight mode, power requirement and the state of both engines.

This provides a greater level of awareness than a warning light. Normally, a human would have to determine the appropriate action on their own, based on the information available (warning lights, error codes, vibrations, etc.). However, if the human (or a decision-making system) receives incorrect or incomplete information they may take an unnecessarily cautious approach and, for example, shut the engine down, or they may continue the current operation failing to take any remedial action. Both of these circumstances can lead to catastrophic consequences

Often an overly sensitive failure detection system can cause “false positive” warnings, i.e., generating an alert for a non-existent fault. This problem is highlighted in a recent Flight International magazine article on the introduction of a new-generation airliner with sophisticated fault detection and alert system (Anon (2010)). One airline experienced a plethora of system nuisance warnings, which: “are driving down technical dispatch (reliability)”. Another operator reported: “What we are grappling with are algorithms for failure detection, which not only detect a failure but also act upon it. Unfortunately this can lead to a perfectly healthy system being shut down or [a no-go fault warning] for a problem that was minor enough to have been deferred.”

The Autonomous PHM system discussed in this paper aims to apply reasoning equivalent to that of a human crew and thus act like an artificial assistant. Such a system could greatly reduce the crew or operator workload in high stress situations, leading to improved levels of safety. This paper uses results of work on the development of PHM and contingency management integrated with autonomous decision-making carried out as a core part of the UK National ASTRAEA Unmanned Air System (UAS) program. This program is paving the way for commercial UAS to operate autonomously in non-segregated airspace within the next decade.

It proposes the integration of PHM into a system at the design stage, based on a PHM Cycle that combines the Design and Operational perspectives. Combining capabilities of current commercial software tools, such as JACK - the autonomous software platform and MADe - the Maintenance Aware Design environment a PHM system is designed offering greater accuracy in the detection of faults, and providing selection of the best response actions ((Glover, W., Cross, J., Lucas, A., Stecki, C., & Stecki, J. (2010)).

2. THE PHM CYCLE

The proposed PHM Cycle is divided into two parts, covering the design and operation of the system as shown in Fig. 1.

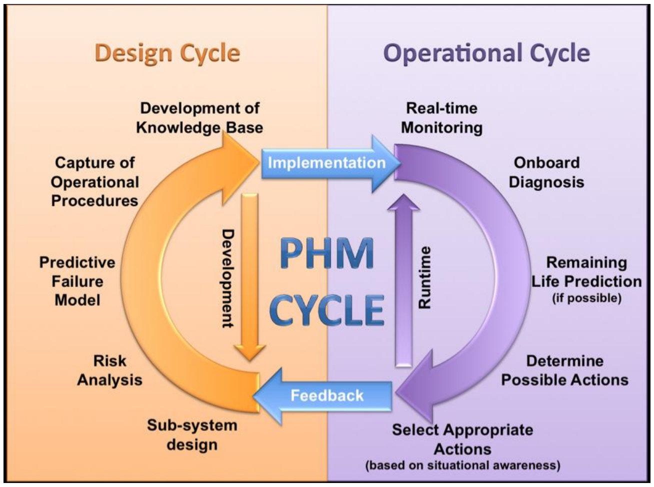
Design Cycle
Operational Cycle
Development of Knowledge Base
Real-time Monitoring
Capture of Operational Procedures
Implementation
Onboard Diagnosis
Development
PHM CYCLE
Runtime
Predictive Failure Model
Remaining Life Prediction (if possible)
Risk Analysis
Determine Possible Actions
Feedback
Sub-system design
Select Appropriate Actions (based on situational awareness)

Figure 1. PHM Cycle

The Design Cycle applies multiple iterations of risk analysis techniques, failure mode prediction, and identification of responding actions to achieve an appropriate level of functional failure coverage. The outcome of this is a knowledge base which can then be applied to a system in operation. The Operational Cycle describes the PHM process when the system is put into operation. It describes how information about faults is gathered, assessed and presented to the end user, or addressed by the autonomous system.

By structuring the PHM design process appropriately, data from the Operational Cycle can be fed back and incorporated into the Design Cycle, yielding continuous improvement in future upgrades or revisions.

2.1 The PHM Design Cycle

The objective of PHM Design Cycle is develop an advisory system which will assess, in real-time, the health of the system and recommends corrective actions to a higher-level decision maker that has to deal with a number of potentially conflicting goals, hostile situations and opportunities apart from input from the PHM. The decision-maker, either a human or a fully autonomous decision system, will have the situational awareness to apply the recommended actions appropriately.

The Design Cycle begins with the specification of the system to be built, which is modeled as a functional block diagram.

Risk Analysis and Determination of Functional Failure Modes. The first requirement of the risk analysis is to identify the possible Functional Failure Modes (FFMs) for the system and to understand their failure dependencies throughout the system. FFMs are the result of specific underlying physical failures triggered by design, manufacturing, environmental, operational and maintenance causes. Such causes (e.g. vibration) can initiate failure mechanisms (e.g. high cycle fatigue) that lead to a fault (e.g. fracture).

The second requirement is to determine how the failures propagate through the system (known as the propagation path) and how this impacts the system functionality. The availability of such information is a key requirement for designing, developing, verifying and validating PHM system design.

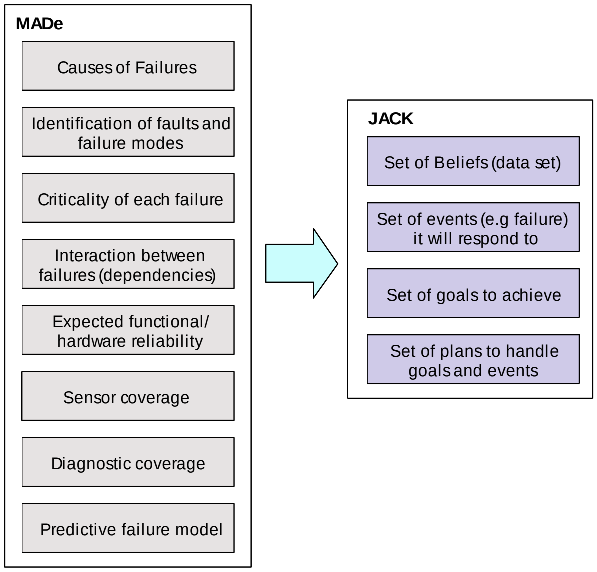
MADe
Causes of Failures
Identification of faults and failure modes
Criticality of each failure
Interaction between failures (dependencies)
Expected functional/ hardware reliability
Sensor coverage
Diagnostic coverage
Predictive failure model
JACK
Set of Beliefs (data set)
Set of events (e.g failure) it will respond to
Set of goals to achieve
Set of plans to handle goals and events

Figure 2. Design of APHM

The outputs of the risk analysis process are usually captured in a Failure Modes and Effects Analysis (FMEA). Once the FMEA is available, the criticality of each FFM is established taking into consideration each specific failure and its propagation paths, the output of this process is the Failure Modes Effects and Criticality Analysis (FMECA) report.

Further assessment of the risk is obtained by carrying out reliability analysis using Reliability Block Diagrams and Fault Trees. Extensive evaluation of system sustainability is conducted using a Reliability Centered Maintenance (RCM) methodology. Reliability analysis is usually performed on the basis of the expected Mean Time Between Failure (MTBF) of hardware components as provided by manufacturers or on the basis of published MTBF standards. In addition to this information PHM requires an assessment of the reliability of specific functional outputs in the system – ‘functional reliability’.

At the conclusion of the risk assessment process, the user can expect to know:

- how the system elements can fail (failure modes)
- the criticality of each failure
- the likely causes of functional failures
- the interactions between functional failures
- what physical failures are linked to functional failures
- the expected functional and hardware reliability of the system.

The information obtained during development of FMECA and Reliability studies is a basis for selecting sensor sets able to detect identified failures and formulating diagnostic rules. This process is discussed in the following section.

There are two type of approaches to the failure risk analysis. The first is a “committee approach” where a team subject matter experts determine failures and their dependencies and subsequent list them in the “spreadsheet” type software. Quality of the analysis depends on knowledge and experience of team members. Reliability studies are usually carried out by a different team of people using specialized reliability software. Sensor selection and development of diagnostic rules cannot directly use the results of FMECA analysis.

The second, model-based risk assessment approach uses existing failure databases and expert knowledge captured in the form of Failure Diagrams and Functional Block diagrams of a system. A standardized functional and failure taxonomy ensures consistency in the interpretation of failure analysis results (Rudov-Clark, S. J., & Stecki, J. (2009)). Reliability models are automatically generated from the functional model of the system. Sensor selection and diagnostic rules are also determined based on automated analysis of the functional model.

Risk assessment as briefly described above forms the basis for any further work on the development of PHM system. Some common problems causing sub-optimal operation of PHM systems can be traced to following risk assessment deficiencies:

- dependencies of failures are not
- identified
- inadequate identification of risks
- incomplete database of failures
- i
  nconsistent language used to define functions and failure concepts
- confusing hardware reliability with functional reliability

different models for Criticality and Reliability Assessments.To overcome these deficiencies MADe - the Maintenance Aware Design environment was used as a risk assessment tool facilitating failure modes analysis and reliability assessment.

Sensor and Diagnostic coverage. Detection of a failure mode is the first and most important step in the PHM process. After all, if we cannot identify failure mode we cannot propose a corrective action. When a failure mode is isolated the reasoning system will attempt to identify the causes of the failure mode.

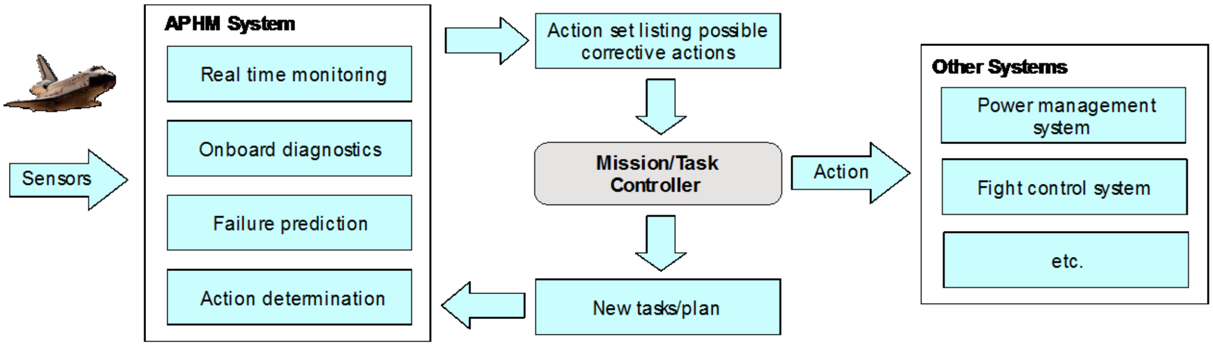

Figure 3. Operational cycle

Sensors are usually selected to detect specific identified failures (e.g. temperature sensor detect temperature change indicating a failure of the heater) thus they are selected to detect symptoms of failures. The sensors are usually selected by personnel responsible for individual components or subsystems who may have only limited knowledge of the impact of their failures on system failures. The final composition of sensor set is decided upon by the system integrator using criteria such as cost, weight, reliability and computing requirements. The overall coverage of system failures is determined using testability analysis software. The diagnostic rules are developed on the basis of symptoms. This methodology has following weaknesses:

- sensor fusion is not based on failure dependencies (fallback – testability)
- diagnostic rules are not based on failure dependencies
- failure coverage is often incomplete and cannot be assessed
- sensor selection does not consider the criticality of failures, or the functional and hardware reliability
- sensor fusion is difficult to implement without failure dependency information.

Model-based approach to sensor selection disposes of some of these weaknesses. The MADe/PHM module uses the model of the system and failure dependencies data obtained in the risk analysis phase and provides the user with an automated 'sensor set design' function (Rudov-Clark, S. D. , Ryan, A. J. ,Stecki, C. M. & Stecki, J. S. (2009)). Each potential sensor set provides a logical cover of the identified failures. In contrast to the above mentioned 'symptom of failure' methodology, the sensor set fuses sensors reading to provide a syndrome of failure. The selection of component/subsystem sensors solely on the basis of failure symptoms can also be carried out and fused with sensor sets based on identification of the syndrome of failure.

By applying this automated approach, with associated capability to conduct trade-off studies of sensor properties such as cost, weight, coverage and reliability, the engineer can select the best possible arrangement of sensors for the given constraints, providing the highest practical level of fault coverage achievable.

Although full coverage of faults is always preferable, it is not necessarily achievable due to system constraints. Also, some failure modes may have degrees of criticality that are below the level of concern and thus they can be excluded from further analysis.

If full failure coverage is not achieved by the set of diagnostic sensors then ambiguity groups exist, i.e. a number of different failure modes have the same system functional responses. These ambiguity groups can be resolved by identifying the most likely fault based on the probability of failure and information about the physical processes and symptoms for each failure provided in the failures database.

The system designer must be aware of the potential implications of any unresolved ambiguities. These ambiguities will directly impact upon the ability of the PHM function to take the best remedial action – if it is unable to identify the correct failure mode then it is unlikely to respond correctly. As such, the designer should, possibly during subsequent design iterations, attempt to remove these ambiguities wherever possible or have contingencies built into the responses to handle their occurrence, for example by integrating BITs or other sensors associated with components.

It is important to remember that the above sensor requirements analyzes are based on a functional model that is qualitative in nature. Thus further quantitative analysis of the sensor set should be considered to validate the results.

The selected sensor set and results of the failure modes and effects analysis provide the basis for the design diagnostic rules needed to identify each failure mode.

Detection and Diagnosis. In on-line, real-time operation inaccurate sensor readings may introduce response patterns which do not correspond to any of the diagnostic rules. One potential solution is the use of multiple redundant sensors that provide a means for resolving differences (e.g. by "voting"). Another solution is the application of reasoning techniques that look for the probable cause of any undefined sensor readings.

Theoretically a sensor set which provides required diagnostic coverage of failure modes will identify all the failure modes. In practice it is not always so. In practical terms a diagnostic sensor set has a certain Probability of Detection (POD) which is a function of reliability of detection, processing, and interpretation of information provided by individual sensors in a diagnostic set. Each failure mode may have different POD. Thus to diagnose a failure mode the reasoning system must not only identify the appropriate sensor responses but also consider Probability of Detection of this sensor and that of a whole sensor set. For example, a pressure sensor may have much higher POD than a vibration sensor, mainly due to the low reliability of vibration signal interpretation.

As the PHM system should provide predictive capability, the failure models should be extended to include information such as historic data of previous failures, results of tests, physics of failure, and length of time for a failure to develop. The length of time it takes for a failure mode to develop, from the initiation of a failure mechanism to the development of a fault and propagation of the subsequent functional failure, is important information for choosing the best actions to mitigate the failure. If a failure is instantaneous, for example, fan blade failure due to catastrophic Foreign Object Damage (FOD), then immediate action will be required. If a failure is gradual there could be some time to perform other actions to slow down the progression of the fault or mitigate its consequences.

For example, compressor blade damage from a bird strike that leads to high-cycle fatigue failure can be addressed by reducing the engine speed thus reducing the rate of crack propagation.

Different faults and failure modes may occur in rapid succession, leading to multiple simultaneous responses being detected.

Developing the Knowledge Base. The knowledge base developed during the PHM Design Cycle includes:

1. a rule base for performing diagnostics and identifying each FFM along with its underlying causes
2. a predicted failure model
3. a set of actions corresponding to each failure.

The knowledge base is designed in such a way that a decision-making system such as an artificial agent can reason about it. If possible, the actions should provide complete coverage of all identifiable failures, and give all possible responses (or actions to be taken) for the identified failure.

With the possible FFM identified, and the sensors chosen and the rules for identifying these failures deduced, the action required for each failure are determined. On-line PHM systems reason about actions in often rapidly changing environments, and operate autonomously. Architectures such as the Beliefs, Desires, Intentions (BDI) model have been developed to deal with these kinds of situations and are implemented in the JACK autonomous software platform.

A JACK agent is a software component that can exhibit reasoning behaviour under both pro-active (goal directed) and reactive (event driven) stimuli. Each agent has:

- a set of beliefs about the world (its data set)
- a set of events that it will respond to
- a set of goals that it may desire to achieve (either at the request of an external agent, as a consequence of an event, or when one or more of its beliefs change), and
- a set of plans that describe how it can handle the goals or events that may arise.

In particular, each agent is able to exhibit the following properties associated with rational behaviour:

- Goal-directed focus – the agent focuses on the objective and not the method chosen to achieve it
- Real-time context sensitivity – the agent will keep track of which options are applicable at each given moment, and make decisions about what to try and retry based on present conditions
- Real-time validation of approach – the agent will ensure that a chosen course of action is pursued only for as long as certain maintenance conditions continue to be true
- Concurrency – the agent system is multi-threaded. If new goals and events arise, the agent will be able to prioritise between them, resolve potential conflicts (e.g. by deliberate to reject or ignore certain goals or delaying their resolution to a later time), and multi-task as required.

When an agent is instantiated in a system, it will wait until it is given a goal to achieve or experiences an event that it must respond to. When such a goal or event arises, it determines what course of action it will take. If the agent already believes that the goal or event has been handled (as may happen when it is asked to do something that it believes has already been achieved), it does nothing. Otherwise, it looks through its plans to find those that are relevant to the request and applicable to the situation. If it has any problems executing this plan, it looks for others that might apply and keeps cycling through its alternatives until it succeeds or all alternatives are exhausted. The BDI agent-is able to be programmed to execute these plans just as a rational person would.

2.2. The PHM Operational Cycle

Once the Design Cycle has been completed and the Autonomous PHM system contains a sufficient level of coverage the system, along with the knowledge base developed, it can be put into use on board of the host system.

The Operational cycle consists of following activities, Fig. 3:

Real-time Monitoring. In operation, the PHM function will receive signals from each of the sensors located in the system or its sub-components. These signals will be constantly monitored, as in conventional systems, so that signal levels that are outside the normal range are detected as anomalies. This differs from conventional approaches as instead of giving a simple warning the anomalies are passed to an on-board diagnostic unit that can provide a response appropriate in the current circumstances, and also show how to reduce or mitigate the identified fault's effects.

On-board Diagnostics. The on board diagnostic unit will make use of the knowledge base developed in the Design Cycle to associate the anomaly or anomalies with a particular FFM. The knowledge base can also provide enough information to identify or predict which physical parts or failure mechanisms are responsible for the failure. If the sensor readings are not sufficient, the diagnostic unit should once again examine reliability data, criticality, and dependencies to determine the FFM. Context specific confirmation rules, can also be applied to help resolve ambiguities or probe further.

Failure Prediction. Once the particular FFM has been identified, the PHM system must predict the remaining life associated with that failure. The failure models (contained in the knowledge base) for the sub-components or parts identified to have failed, will be analyzed in order to determine what time constraints are involved and how the failure will develop.

Action Determination. The PHM system now has all the information it needs to make an informed decision about which actions it should take (in the case of an autonomous system), or recommend. It now has at its disposal:

- the sensor readings perceived to be anomalous
- the functional fault this corresponds to
- the physical defect or failure likely to have caused this fault
- a model of how the system will continue to fail, including the estimated time before further failures occur. Using the above information the PHM system will select the actions that it perceives to be the best for the given situation.

Depending on application, PHM capability can be designed into autonomous or semi-autonomous systems to diagnose faults, predict remaining functional life and suggest reasonable actions to deal with these events, if (or when) they occur. When deployed, depending on application, the action determined by Autonomous PHM would not necessarily be the final action to be performed. This is due to the Autonomous PHM not necessarily having complete knowledge of the situational context surrounding the system's operation. In such application it would pass the appropriate action alternatives to a higher-level decision-making system or human user who, in turn, would make this selection and initiate the associated action.

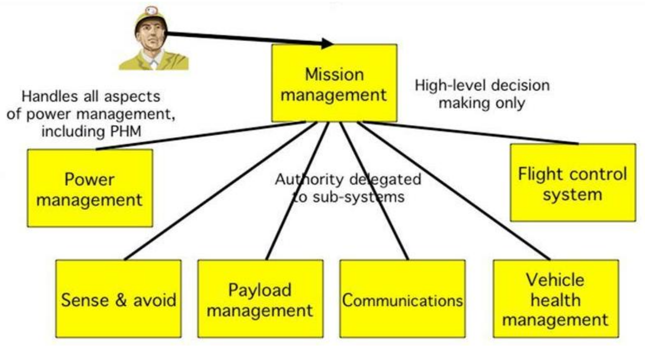
Mission management
Handles all aspects of power management, including PHM
High-level decision making only
Power management
Authority delegated to sub-systems
Flight control system
Sense & avoid
Payload management
Communications
Vehicle health management

Figure 4. Delegated Autonomy Architecture

3. EXAMPLE: POWER MANAGEMENT ON A UAV

A typical example is an autonomous, real-time, engine health and power management on an Unmanned Air Vehicle (UAV) where it might manage the specific subsystems (i.e., the engine, drive trains, etc.) of the overall vehicle.

The PHM and Power Management, Fig. 4, forms part of a delegated autonomy architecture in an autonomous system with the human overseer always remaining in the position of ultimate management responsibility. It will not know how critical these requirements are with respect to the overall task being performed by the vehicle it is attached to. It is the responsibility of the high-level decision maker to evaluate the mission or task, as it is in the best position to make such a decision. It can then feed new requirements to the Autonomous PHM.

Consider a UAV in flight: the autonomous software must be able to handle faults when they occur with equivalent or better levels of competence than a human pilot if the UAV is to achieve civil certification. The faults identified may require actions to be taken to avert danger and could cause the mission to be altered or abandoned.

Design. The example being used is the lubrication system on the Rolls-Royce 250 engine, and how failures can occur, e.g. of bearings. The FMECA analysis was completed in MADe. The autonomous PHM capability is being implemented in AOS's C-BDI, and the operational scenario is based upon a twin-engine UAS operating at high power in a hot and high altitude environment. It is expected that this demonstrator will be completed in 2012 and the results published at that time.

1. The development of PHM system followed the above Design Cycle methodology:a functional model was created of the engines, including the interactions between the critical internal components (over 12000 functional connections).a risk analysis was performed determining the various ways the engine can fail. The sensor types and locations are chosen and rules identified that associate the various sensor readings to FFMs. Data would be included from previous applications of that engine type or similar engines, such as maintenance logs, failure rates, and results of examinations performed on previously failed engines.
4. the reliability data, when available, will be used to aid in the creation of the failure models
5. the agent actions are under constructions taking into account all of the possible actions that can be done to the engine. These may include increasing or decreasing the thrust or shutting down the engine completely
6. the knowledge base is being created to be inserted into the PHM function on the UAV.

Operation. Consider the scenario of the UAV performing a search and rescue mission. During the operation a bearing within an oil pump on one of the two engines begins to suffer from wear.

The PHM system would monitor the engine sensors, detect any anomalies, and determine if these are significant (e.g., not just a spike due to a power on/off transition). The FFM would be detected by sensors as a loss of oil pressure within that engine which, when compared to the diagnostics rules contained within the knowledge base would indicate a pump failure. By examining the failure probability of each component within the pump, the level of functionality lost, and the rate at which functionality is decreasing, the power management system would recognize that the cause is likely to be bearing failure.

Analysis of the failure model for bearing wear failure will give the probable lead-on effects of this failure mechanism. The system would then examine the possible actions to overcome this failure, which may include:

- shutting the engine down immediately;
- reducing thrust to 60% before continuing operation for up to 2 hours;
- reducing thrust to 30% for 4 hours; and,
- other combinations.

The PHM system capability would then assess these actions, based upon the following situational information:the current power requirement is that both engines need to operate at 30% thrust for 2 hours;.due to a fault that occurred earlier, the second engine has already been shut down; andthe remaining engine is currently running at 80% thrust to compensate.

For the given situation the PHM would recommend the following actions:turn on the second engine, and operate both engines at 30%, possibly damaging the second engine further;leave second engine shut down and reduce thrust as much as possible, however it must be at least 60% to meet the power requirements;abort or alter the mission since the power requirements cannot be met; orreduce thrust to 70% and see if the oil pressure returns to nominal level. If it does, continue with the engine power at that level, otherwise reduce further.

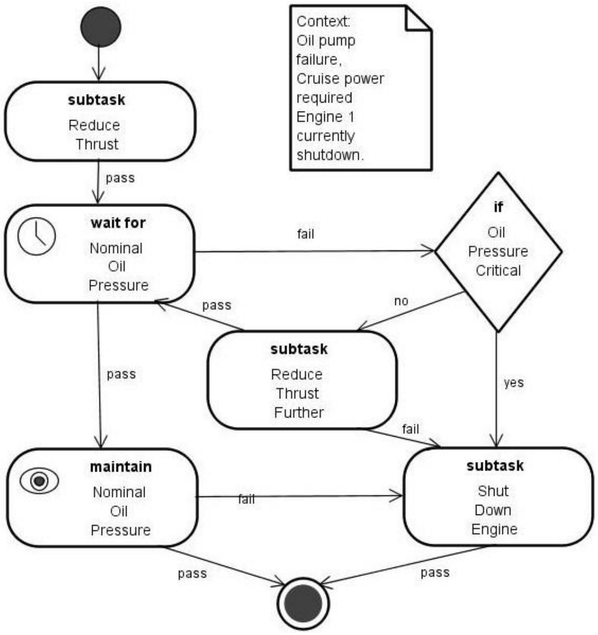

Figure 5. JACK Plan to Handle an Engine Fault.

An example of a JACK graphical plan that implements this is shown in Fig. 5. This shows how after reducing thrust the oil pressure will be monitored for some time to see if the problem is mitigated (the *wait_for* block). If it is not then the thrust is reduced further. If the problem gets worse, then the engine is shut down. If the problem is mitigated, the maintain block will keep monitoring the problem to make sure it doesn't get worse in the future.

Upon receiving these possible actions, the higher-level decision-making software can determine if the mission is important enough to continue (at the risk of further failure) or if it can be altered. Instead of being overloaded with multiple options, or receiving insufficient information from multiple simple warnings, the autonomous system will receive a set of possible actions that are succinct and meaningful. From this set it can choose the best action for the given situation.

4. Discussion and Conclusions

This functional failure mode approach, based on using reasoning to improve the diagnosis will maximize the likelihood of determining the failure mode correctly, and determine the most appropriate course of action – taking into account current circumstances (e.g., flight mode, power requirement and the state of both engines). Autonomous systems must have this capability to operate successfully. Manned systems will also benefit by improving the accuracy of failure mode identification and recommending the best action to take. By acting like an artificial assistant, such a system could greatly reduce the crew or operator workload in high stress situations, leading to improved levels of safety.

By structuring the PHM design process appropriately, data from the Operational Cycle can be fed back and incorporated into the Design Cycle, yielding continuous improvement in future upgrades or revisions of the UAV.

The novelty of the system presented here derives from the combination of a risk assessment tool with the high-level representation and flexibility offered by a decision-support tool, making the resulting system appropriate for integration into a complex architecture for autonomous vehicles where multiple levels of delegation and decisions (possibly including the human) interact to determine and adapt the course of actions during a mission.

ACKNOWLEDGEMENT

The authors would like to thank the Technology Strategy Board (TSB) for funding the ASTRAEA and ASTRAEA II programs which make work like this possible.

REFERENCES

Anon (2010). A380 In-service report. *http://www.flightglobal.com/page/A380-In-Service-Report/Airbus-A380-In-Service-Technical-issues.*

Glover, W., Cross, J., Lucas, A., Stecki, C., & Stecki, J. (2010). The Use of Prognostic Health Management for Autonomous Unmanned Air Systems, *Proceeding of International Conference on Prognostics and Health Management,* October 10-16, Portland, Oregon, USA

Kurtoglu, T., Johnson, S. B., Barszcz, E., Johnson, J.R., Robinson, P. I. (2008). Integrating System Health Management into the Early Design of Aerospace Systems Using Functional Fault Analysis. *Proceeding of International Conference on Prognostics and Health Management,* Oct 6-9, Denver, Colorado, USA.

Rudov-Clark, S. J., & Stecki, J. (2009). The language of FMEA: on the effective use and reuse of FMEA data. *Sixth DSTO International Conference on Health & Usage Monitoring*. March 9-12, Melbourne, Australia.

Rudov-Clark, S. D. , Ryan, A. J. ,Stecki, C. M. , Stecki, J. S. (2009). Automated design and optimisation of sensor sets for Condition-Based Monitoring. *Sixth DSTO International Conference on Health & Usage Monitoring*. March 9-12, Melbourne, Australia.

Scheuren, W. J., Caldwell, K. A., Goodman, G. A. and Wegman, A. K. (1998). Joint Strike Fighter Prognostics and Health Management. *Proceedings of the 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit.* July 13-15 1998, Arlington