The Use of Prognostic Health Management for Autonomous Unmanned Air Systems

William Glover[1], Joshua Cross[1], Andrew Lucas[1], Chris Stecki[2], and Jacek Stecki[2]

[1]*Agent Oriented Software Limited, Cambridge CB1 1BH, United Kingdom*
william.glover@aosgrp.co.uk
joshua.cross@aosgrp.com.au
andrew.lucas@aosgrp.co.uk

[2]*PHM Technology, Melbourne, Victoria, 3170, Australia*
jstecki@phmtechnology.com
cstecki@phmtechnology.com

ABSTRACT

Unmanned Air Systems (UAS) show great promise for a range of civilian applications, especially 'dull, dirty or dangerous' missions such as air-sea rescue, coastal and border surveillance, fisheries protection and disaster relief. As the demand for autonomy increases, the importance of correctly identifying and responding to faults becomes more apparent, as fully autonomous systems must base their decisions solely upon the sensors readings they receive – as there is no human on board. A UAS must be capable of performing all the functions that would be expected from a human pilot, including reasoning about faults and making decisions about how to best mitigate their consequences, given the larger context of the overall mission. As these autonomous techniques are developed their benefits can also be realised in non-autonomous systems, as real-time aids to human operators or crew. This paper proposes a novel approach to PHM that combines advanced Functional Failure Mode Analysis with a reasoning system, to provide effective PHM for autonomous systems and improved diagnosis capability for manned aircraft.*

1. INTRODUCTION

This paper describes how 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.

The objective of this paper is to show how PHM concepts can be included in system design and in doing so, provide effective prognostic capabilities from within the system in operation. This has far greater benefit than a PHM system that is 'attached to' or added on top of an existing piece of hardware (Kurtoglu, et. al., 2008). This paper describes the integration of PHM into a system at the design stage, based on a PHM Cycle that includes both the Design and Operational perspectives. Making use of current commercial software tools such as the JACK autonomous software platform – 'JACK' (Busetta, Rönnquist, Hodgson, & Lucas, 1999); and the Maintenance Aware Design environment – 'MADe' (Rudov-Clark & Stecki, 2009), a greater accuracy in detection of faults can be achieved, and selection of the best response actions can be provided. These advantages are revealed when examining the potential application of these PHM concepts to engine health and power management on an Unmanned Air System (UAS).

Programmes such as ASTRAEA (Technology Strategy Board, 2010), are paving the way for commercial UASs to operate autonomously in non-segregated airspace within the next decade, including the development of PHM and contingency management integrated with autonomous decision-making. This capability has already been demonstrated in practice, with flight trials of the UK QinetiQ/Ministry of Defence BAC 1-11 Autonomous UAV Surrogate in 2007 and the forthcoming flight trials of the UK Ministry of Defence/Industry *Taranis* Unmanned Combat Air Vehicle demonstrator (Lucas, et. al., 2010).

2. THE "PHM CYCLE"

The authors propose a PHM Cycle that is divided into two parts covering the design and operation of the system (See Figure 1).

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 fault 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

2.1.1 System Design

The Design Cycle begins with the specification of the system to be built, which is modelled as a functional block diagram. MADe, a software tool, enables an engineer to create functional models from the initial requirements and specifications of the system. The models make use of generic components or sub-systems provided by MADe. These can subsequently be augmented or modified with specific data as the design matures.

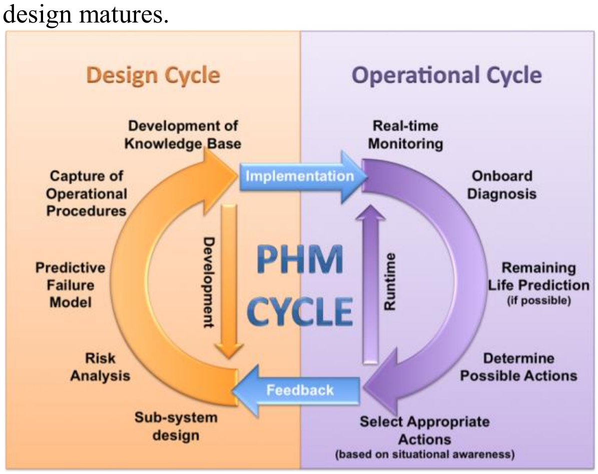

Figure 1, PHM Cycle

2.1.2 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 dependency flows throughout the system. FFMs are the result of specific underlying physical failures triggered by design, manufacturing, environmental, and maintenance causes. Such causes can initiate failure mechanisms (e.g. fatigue) that lead to a fault (e.g. fracture).

Often systems are designed and the PHM analysis focuses on component failures, i.e., the characteristics of these physical failures, rather than their impact on the functionality of the system.

MADe automates the generation of the dependency mapping required to determine end effects within the system or systems. It generates a database of specific system responses to each FFM, linked automatically to physical causes or mechanisms of failures and faults. The availability of such information is a key requirement for designing, developing, verifying and validating PHM system design.

The outputs of the risk analysis process are usually captured in a Failure Modes and Effects Analysis (FMEA). To ensure the consistency and accuracy of the system model and to optimise its extensibility to other applications, MADe failure concepts (cause, mechanism, fault & failure modes) are generated in a standardised taxonomy. Once the FMEA is available, the criticality of each FFM is established taking into consideration each specific failure and its 'propagation paths' (the consequence of failure on the functional performance of the system). MADe's output of this process is the Failure Modes Effects and Criticality Analysis (FMECA).

Further assessment of the risk is obtained by carrying out reliability analysis. Conventional reliability analysis is 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). However PHM requires, in addition to hardware reliability assessment, an assessment of the reliability of specific functional outputs in the system – 'functional reliability'. Using the functional model as the basis, MADe provides the user with both hardware (component) and functional reliability assessments that provide the data for availability and maintainability assessments.

At the end of the risk assessment process, the user has knowledge of:

- what the causes of failures are;
- how the system can fail;
- how critical each failure is;
- what the interaction between failures is; and,
- what the expected functional and hardware reliability of the system is.

This data allows MADe to identify the required set of sensors to provide appropriate coverage of critical faults and FFMs. MADe uses the FMECA data to both determine the key FFMs that require diagnostic coverage and to validate the testability of the sensor set.

Although 100% coverage of faults is always preferable, it is not necessarily achievable due to factors including the cost and weight of the sensors, the practicality of the required sensors (physical dimensions, accessibility), and their reliability. Some of the failure modes may have degrees of criticality that are below the level of concern and thus they could be excluded from further analysis.

MADe provides the user with an automated 'sensor set design' function and the capability to conduct trade studies of the sensor sets based on user defined parameters (cost, weight, coverage, etc.). The sensor sets generated by MADe and dependencies analysis provide the basis for the design diagnostic rules needed to determine each failure mode. By applying this automated approach the engineer can select the best possible arrangement of sensors for the given constraints, providing the highest practical level of fault coverage achievable.

If 100% fault coverage is not achieved by the set of diagnostic sensors, i.e. if ambiguity groups are present (the same system dependency response for different failure modes), these must be resolved. This can be achieved by identifying the most likely fault by correlating the sensor readings with reliability data, the criticality of components and knowledge of the failure dependencies.

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.

One other source of potential error is inaccurate sensor readings, which will invalidate the diagnostic rules. One example of an in-flight sensor error is highlighted in the loss of the B2 on take-off at Guam (Anon 2008), where the pitot static system generated erroneous airspeed readings. One approach to solve this is multiple redundant sensors that provide a means for resolving difference (e.g. by "voting"). The authors are looking at using deliberative reasoning techniques to complement this approach. For example a pilot will often feel the extension of the flaps on the motion of the aircraft, and so detect if they extended or not – he or she reasons that there is an inconsistency in the aircraft and then looks for the cause. In a similar manner it is proposed that intelligent pre-processing of the incoming PHM data can offer a similar degree of redundancy.

2.1.3 Predictive Failure Model Development

A failure model should be produced that describes for each fault (or possible fault): how the fault will propagate through the system; other induced faults; and the length of time it will take to propagate. This may require further modelling of the system's subcomponents and physical parts. The failure models will be based upon knowledge of existing parts, including autopsies from previous failures and individual part tests and evaluations. These failure models can come from a range of techniques and tools. MADe has the role of bringing these together in a form of a coherent summary, relating symptoms to functional faults and the physical causes.

The length of time it takes for a fault to propagate is vital information for choosing the best actions to mitigate the fault. If a failure is instantaneous (e.g., a fan blade failure due to catastrophic Foreign Object Damage (FOD)) then immediate action will be required, otherwise there could be some time to perform other actions to slow down the progress of the fault or mitigate its consequences (e.g., compressor blade damage from a bird strike leading to high-cycle fatigue failure, reducing engine speed will reduce the rate of crack propagation).

Potentially, different failures and failure modes may be identified rapidly in succession, leading to a significant number existing concurrently. In this case, the PHM function will have to reduce the scope of its model and only consider failures that are imminent or critical.

2.1.4 Capture of Operational Procedures

With the possible FFM identified, the sensors chosen and the rules for identifying these failures deduced, the action (or actions) required for each failure must be determined. Traditionally these recommended actions have been captured in a textual form such as an operational handbook and checklist. However this relies on the user having the time to look up these actions or memorise them and then have the situational awareness to apply them appropriately. By determining the actions required for each FFM the user can be prompted with such actions, rather than relying on a manual or being left to identify and follow a check list.

Autonomous systems must be able to make their own decisions and thus cannot rely on user input or paper manuals. In semi-autonomous systems recommendations must be passed to higher-level decision-making systems, whether these are human or machine. In both cases, these require the action sets to be known to the system in order for it to make or aid in the decision making process.

The level of autonomy of a UAS subsystem can be described by the Pilot Authority and Control of Tasks

(PACT) Taxonomy (Taylor, Abdi, Drury, & Bonner, 2001) which was developed as part of the UK MoD's Cognitive Cockpit (COGPit) Programme. This taxonomy is shown in Figure 2 and describes the contract between the pilot and autonomous subsystems. With a high PACT Level the autonomous systems have more control over decision-making and the pilot has less workload. Conversely at lower PACT levels the autonomous decision-making systems can only give advice and the pilot remains in full control of the vehicle. This is a model of how intelligent PHM could be used in an autonomous system or for decision support in a non-autonomous system.

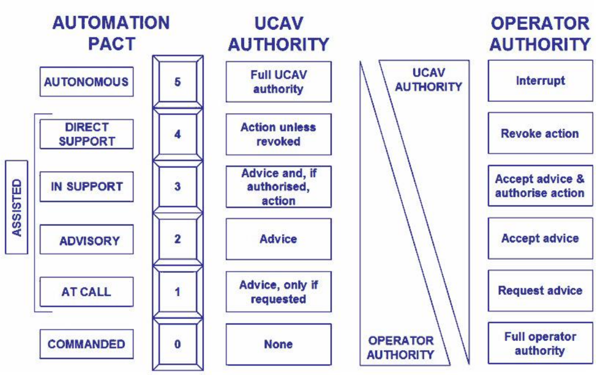

Figure 2, PACT Levels of Autonomy

PHM requires reasoning about actions in rapidly changing environments, as such it is inappropriate to use conventional expert systems, which are not suited to time varying (non-monotonic) domains and lack a sophisticated knowledge representation. This has led to work on reasoning about actions, and theories of agency (Bratman, 1987). Architectures such as the Beliefs, Desires, Intentions (BDI) model have been developed to deal with these kinds of situations. (Rao & Georgeff, 1991).

BDI agents were first implemented in PRS (Procedural Reasoning System) as a system for automating and controlling spacecraft systems, in particular, the handling of malfunctions in the Reaction Control System (RCS) of NASA's space shuttle (Georgeff & Ingrand, 1989) and (Ingrand, Georgeff and Rao, 1992). PRS was used because of its ability to reason about and perform complex tasks in a flexible and robust manner, somewhat like a human assistant. PRS used goal-directed reasoning whilst remaining reactive to unanticipated changes in its environment.

Continuing the PRS lineage of BDI systems are dMARS and JACK (Busetta, Rönnquist, Hodgson, & Lucas, 1999) which are distinguished by their intuitive, ~~SME~~ Subject Matter Expert (SME)-friendly representation language compared to alternative systems, which can be tedious to build and difficult for SMEs to understand.

2.1.5 Develop Knowledge Base

The knowledge base developed during the PHM Design Cycle includes:

- a rule base for performing diagnostics and identifying the FFM or the underlying cause;
- a predicted failure model; and,
- a set of actions corresponding to each fault.

The knowledge base should be designed in such a way that a decision-making system such as a BDI agent can reason about it. If possible, the actions should provide complete coverage of all identifiable faults, and give all possible responses (or actions to be taken) for the identified fault. If not, then probabilistic methods, such as Bayesian techniques, may be required to distinguish between the modes.

2.2 The PHM Operational Cycle

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

2.2.1 Real-time Monitoring

In operation, the PHM function will receive signals from each of the sensors located in the system or its subcomponents. 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 onboard 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.

2.2.2 Onboard Diagnostics

The onboard 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.

2.2.3 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 subcomponents or parts identified to have failed, will be analysed in order to determine what time constraints are involved and how the failure will develop.

2.2.4 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; and
- a model of how the system will continue to fail, including the estimated time before further failures occur.

From this information the PHM function will select the actions that it perceives to be the best for the given situation.

When deployed, the PHM function would not select the final action to be performed. Instead it would pass the appropriate alternatives to a higher-level decision-making system or human user who, in turn, would make this selection and initiate the associated action. This is due to the PHM function not necessarily having complete knowledge of the situational context surrounding the system's operation.

For example the PHM function might manage the power systems (i.e., the engine, drive trains, etc.) but this is only one element of the overall vehicle. As in Figure 3 the PHM and Power Management forms part of a delegated autonomy architecture in an autonomous system, such as an unmanned air vehicle (UAV) or unmanned underwater vehicle (UUV). However the human overseer always remains in the position of ultimate management responsibility.

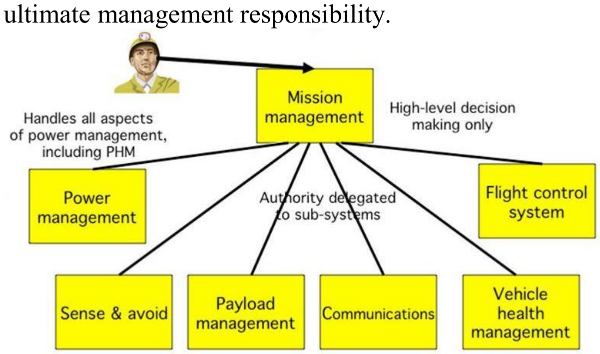

Figure 3 - Delegated autonomy architecture

Hence the two systems work together to take appropriate action in response to a functional fault being detected.

The PHM function will be aware of the current requirements of the system it is managing (i.e. how much power is needed), as well as ~~the~~its current health ~~of the system~~ (i.e~~.,~~. if any previous problems have ~~been identified~~occurred). It will not know how critical ~~this power level is~~these requirements are with respect to the overall task ~~it's performing.~~being performed by the vehicle it is attached to. This is to separate the PHM from the higher-level reasoning. The PHM function should only respond to what is required of it and if it cannot meet these requirements it should inform the high-level decision maker.

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 PHM function.

3. EXAMPLE: POWER MANAGEMENT ON A UAV

Consider a UAS 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 UAS 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.

The following scenario assumes a twin Rolls-Royce C250 engine UAS that has a PHM power management system including a knowledge base covering faults that can occur within these engines.

3.1 Design

The system would be designed as per the above Design Cycle description.

1. A model is created of the engines, including the interactions between the critical internal components.

2. A risk analysis is performed determining the various ways the engine can fail. The sensor types and locations are chosen and rules identified that connect 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. The reliability and criticality date obtained would be used to aid in determining the physical cause of the failure.

3. The reliability data is also used to aid in the creation of the failure models.

4. The action set is created 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.

5. The knowledge base is created and ready to be inserted into the PHM function on the UAS.

3.2 Operation

Consider the scenario of the UAS 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 too much 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 a sensor as a loss of oil pressure within that engine which, when compared to the rules generated by the MADe tool (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 which functionality is decreasing, the power management system would recognise 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 ~~4hrs~~4 hours; and,
- other combinations.

The PHM system 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~~hours;
- Due to a fault that occurred earlier, the second engine has already been shut down; and,
- The remaining engine is currently running at 80% thrust to compensate.

For the given situation the PHM would recommend the following actions:

1. Turn on the second engine, and operate both engines at 30%, possibly damaging the second engine further;

2. Leave second engine shutdown, and reduce thrust as much as possible, however it must be at least 60% to meet the power requirements;

3. Abort or alter the mission since the power requirements cannot be met; or,

4. Reduce 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. An example of a JACK graphical plan that implements this is shown in Figure 4. 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 shutdown. If the problem is mitigated, the *maintain* block will keep monitoring the problem to make sure it doesn't get worse some time 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.

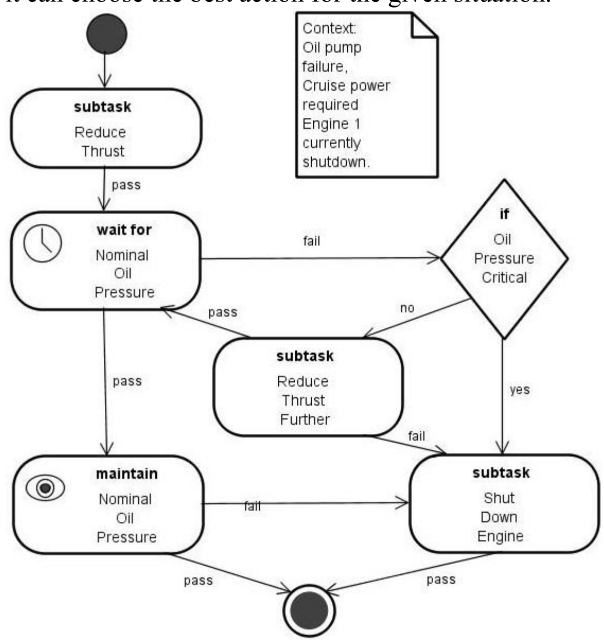

Figure 4 - An example JACK plan that handles an engine fault

4. POTENTIAL BENEFITS

Functional FMECA analysis ensures that functional failures are identified, unlike ad-hoc FMECA analysis, which often mixes functional and physical failures. The FMECA generated by MADe minimizes ambiguity between failure modes where possible by ensuring the optimum combination of sensors, including sensor types, quantities and locations. When this FMECA is combined with existing diagnostic techniques (including probabilistic methods), there is a higher success rate in accurately identifying the FFM and the failure mechanism.

If a predicted remaining life model is available this can then be applied to assist with corrective action, as it allows for decisions to be made based on what is likely to occur rather than purely on what has occurred.

Having identified the functional failure modes and determined their criticality, reasoning techniques can be applied to determine a set of actions that are feasible for the given situation. 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 then they may take an unnecessarily cautious approach and shut the engine down, or may continue the current operation failing to take any remedial action. Both of these circumstances can lead to catastrophic consequences, as illustrated by the following examples.

The consequence of incorrect diagnosis of a fault is dramatically illustrated by the loss of British Midland flight 92 (Anon 1989). After an engine fault the aircrew used incorrect assumptions about the symptoms of failure and mis-diagnosed engine no. 2 as failing. A proper review to check the instruments and review decisions may have exposed the error, however the crew was already overloaded with other tasks and didn't complete this process. An intelligent PHM system would have been able to analyse the malfunction without distraction, offer advice on the correct engine to shut down, or alert the pilots if the wrong action was taken.

Another example is the loss of a Northrop Grumman B2 stealth bomber (Anon 2008). A common point of failure existed for airspeed measurement that caused the Flight Management System (FMS) to stall the aircraft on take-off. A more sophisticated reasoning process would have noticed the airspeed as being anomalous and used other sources of information, such as GPS-derived speed. This would have highlighted the readings given by the airspeed sensor as potentially invalid and could possibly have averted this accident.

The PHM function described can potentially avoid disasters such as these by offering superior detection, diagnosis and procedures that increase the likelihood of the best actions being taken.

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 (Anon 2010) on the introduction of a new-generation airliner with sophisticated fault detection and alert systems. 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 detects a failure but also acts 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 PHM function proposed in this paper aims to eliminate this problem by applying reasoning, equivalent to that of a human crew. The reasoning system is able assess the sensor readings and determine if they are only fluctuations, avoiding unnecessary shutdowns and dispatch delays.

5. DISCUSSION AND CONCLUSION

This functional failure mode approach, based on using reasoning to improve the diagnosis will maximise 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, 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.

Currently the authors are implementing and demonstrating the capabilities of this approach in the ASTRAEA programme. 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 is in progress and is being brought together 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 environment. It is expected that this demonstrator will be completed in 2012 and the results published at that time.

ACKNOWLEDGMENT

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

REFERENCES

Anon (2008). *Forgotten Lesson Caused B-2 Crash.* (2008, Jun 6). From Aviation Week: http://www.aviationweek.com/aw/generic/story_channel.jsp?channel=defense&id=news/B-2060608.xml&headline=Forgotten%20Lesson%20Caused%20B-2%20Crash

Anon (1989). *Report on the accident to Boeing 737-400, G-OBME, near Kegworth, Leicestershire on 8 January 1989.* Department of Transport, Air Accidents Investigation Branch, London.

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

Bratman, M. E. (1987). *Intention, plans, and practical reasoning.* Cambridge, MA, USA: Harvard University Press.

Busetta, P., Rönnquist, R., Hodgson, A., & Lucas, A. (1999). JACK intelligent agents - Components for intelligent agents in JAVA. *AgentLink News Letter , 2*.

Georgeff, M. P., & Ingrand, F. F. (1989). Monitoring and control of spacecraft systems using procedural reasoning. *Proceedings of the Space Operations Automation and Robotics Workshop*.

Ingrand, F. F., Georgeff, M. P. & Rao, A. S. (1992). An Architecture for Real-Time Reasoning and System Control. *IEEE Expert 7(6)*.

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. *International Conference on Prognostics and Health Management*

Lucas C.A., Lyons, A., Glover, W., Cross, J. S. (2010) Incremental Saftey Assurance of Autonomous Decision-Making System for UAS, *IET International System Safety Conference*

Rao, A. S. & Georgeff, M. P. (1991). Modeling Rational Agents within a BDI-Architecture. *Proceedings of the Space Operations Automation and Robotics Workshop* .

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* .

Taylor, R. M., Abdi, S., Drury, R., & Bonner, M. C. (2001). Cognitive Cockpit Systems: Information Requirements Analysis for Pilot Control of Automation. *Engineering psychology and cognitive ergonomics , 5*, 81-88.

Technology Strategy Board (2010), *Technology Strategy Board to Invest Further £4.5 Million to Develop Autonomous Unmanned Air Systems Technology*