

# Health Monitoring for Reconfigurable Integrated Control Systems

**Dr Mark Nicholson**

Department of Computer Science, University of York  
York, England

## Abstract

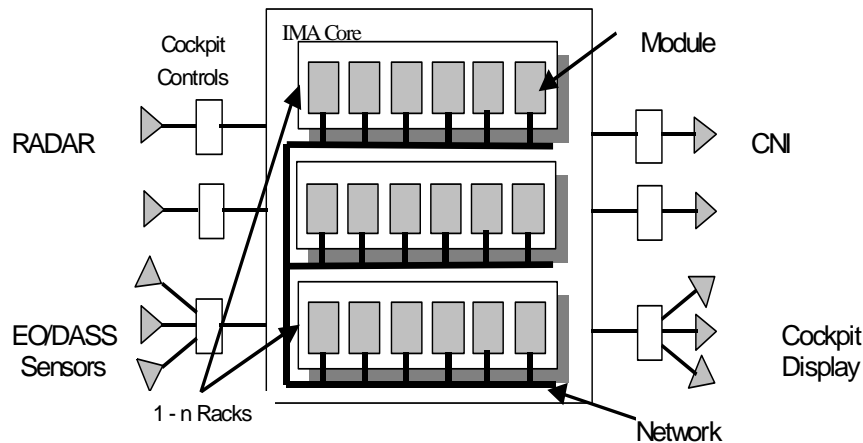
*The next generation of control systems are likely to be characterised by much higher integration, where common / shared computer resources perform multiple system functions. It is possible to reconfigure such systems to provide continued functionality when an element of the system fails. To achieve this aim a number of pre-requisites must be in-place: the ability to determine when a failure has occurred, the appropriate configuration to move to and the ability to safely transfer from one configuration to another. This paper concentrates on the first of these in the form of health monitoring systems for IMS. The approach takes into account the potentially safety critical nature of the applications and the nature of these computer systems.*

## 1. Introduction

Most current control system architectures, such as avionics systems, are federated systems with each function located within its own processor and connected to each other by a data bus. Integrated Modular Avionics (IMA) (EUROCAE 2004) is a term to describe a distributed real-time computer network aboard an aircraft. This network consists of a number of computing modules capable of supporting many applications, which in turn may have different safety criticality levels. One possible IMA architecture is presented in Figure 1 (ASAAC 2002). Each module contains an application that 'services' either a sensor or output or both. A common shared bus network connects the sensors, modules and outputs. Other domains, such as the automotive sector, have also looked at the concept of IMA. Thus, in this paper the more general term Integrated Modular System (IMS) is employed.

Reconfiguration is the capability of a system to adapt its functionality to the changing conditions of its environment (Trapp and Schurmann 2002). One such event could be a change in the mode of operation of the system, such as a move from the initialisation mode to the running mode. Another event that may be addressed via reconfiguration is a failure of one or more elements of the system. This could be a hardware, software or logical failure. This implies that the system has the ability to adapt its behaviour in the presence of faults to achieve continued safe operation and graceful degradation. Limited reconfiguration capability already exists in federated systems but the potential is much greater in IMS. Thus one of the benefits of moving to IMS is the ability to reconfigure the system in response to a range of triggering events.

One current approach to failure management in a federated system is to employ redundancy; that is to employ multiple copies of a system element. In the long term it may be possible to trade-off the level of redundancy employed in a safety related control system with the ability to provide “reconfiguration on failure”.



**Figure 1: ASAAC Architecture for an IMA radar system**

If “reconfiguration on failure” is to provide effective fault-management the ability to determine when a reconfiguration should take place is required. To accomplish this the concept of health monitoring needs to be adapted and extended to take into account the potentially safety critical nature of the applications placed on the IMS platform and the characteristics of IMS computer systems. Health monitoring (HM) is the ability to identify the failure of one or more system elements. Historically, health monitoring has been used to provide maintenance-related failure data for mainly mechanical systems. For instance the F22 flight critical systems have extensive self-diagnostics and built-in testing capability for the various subsystems (Globalsecurity 2004). There are more than 15,000 fault reports available for the avionics systems. Most of these are low-level fault reports that do not result in warnings or degrade the operation of the aircraft.

In IMS health monitoring could be the function responsible for monitoring the system to detect, and report hardware and software (application and operating system) faults and failures. The fault management part of the IMS then uses this information to determine the appropriate system level response, such as reconfiguration. Thus, an ability to detect, and handle failures in such systems become requirements that the system must comply with in order to meet safety objectives. One of the decisions that must be made for instance is which combinations of failure reports will lead to a reconfiguration. Furthermore a decision must then be made as to how extensive the reconfiguration will be. Safety implications accrue if either of these decisions is incorrect.

In Section 2 the concept of a configuration of the elements of an IMS is introduced. Reconfiguration mechanisms are then discussed with reference to the requirement to be able to safely reconfigure a system on failure. The elements of a

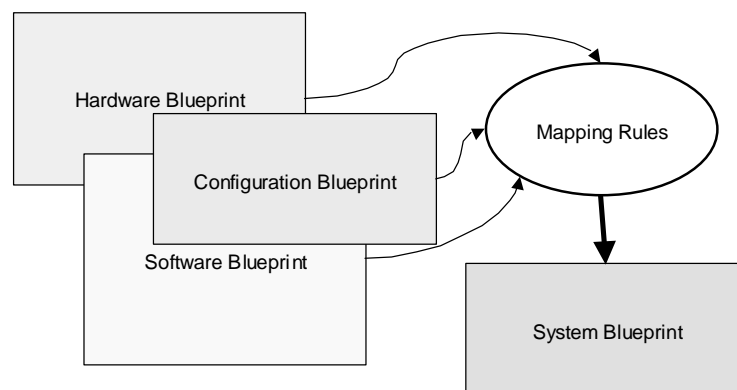
health monitoring system for IMS are presented in Section 3. The paper then introduces possible safety analysis of a proposed reconfiguration on failure mechanism for IMS in Section 4. Finally, the work still needs to be undertaken to extend and validate the approach, for instance to provide safety argument and evidence is presented.

## 2. IMS System Blueprints

### 2.1 Configurations

A configuration of a system consists of a set of hardware elements (sensors, actuators, processors, communication buses, etc) and software elements (applications, operating systems, device drivers, embedded software in the sensors, etc). The configuration is set up to meet a given set of system level requirements such as timing, functionality, computing resource usage and fault tolerance requirements. In an IMS each function capable of being run as software can be mapped to any of a number of processors. Thus, a mapping of the software to the hardware is required (Nicholson, Hollow and McDermid 2000) for a given set of applications to be run on the IMS platform. In some standards, for example ASAAC, this mapping is referred to as a system configuration or “System Blueprint”.

Blueprints can take many forms. Each is a generic template for that part of the system, with its own constraints (e.g. hardware performance limitations). The ‘best’ bits are taken from each blueprint to create the System Blueprint that can then be loaded on to the relevant IMS platform. This will depend upon a set of constraints or ‘rules’. Figure 2 below shows how this is designed to work conceptually.



**Figure 2: Elements of a System Blueprint (Joliffe 2004)**

The elements in Figure 2 are:

**Software Blueprint** – contains a description of each software application in terms of memory, scheduling and communications requirements.

**Hardware Blueprint** – contains a description of each type of hardware employed (sensors, actuators, processors, communication buses, etc). For the computer modules for example this blueprint will contain data on the available memory, processor type and speed, and available communications.

**Configuration Blueprint** – contains a description of how the hardware and applications can be physically and logically connected together e.g. bandwidth, maximum number of connections, etc.

**Mapping Rules** – optimise the Software, Hardware and Configuration blueprints against a set of constraints.

**System Blueprint** – also known as a Run-time Blueprint, is the output from the application of the mapping rules, and can be implemented on a target system or platform.

A number of projects have been working on this issue. The VICTORIA (VICTORIA 2001) project looks at mappings / blueprints for civil aerospace IMS and the ASAAC standards look at blueprints for IMS in the military domain.

## 2.2 Reconfiguration Mechanisms

A single system blueprint can be produced for a system and several methods exist to do this (Bates et al 2003). There are a number of reasons however, why it may be desirable to change the system blueprint at run-time. A change from one system blueprint to another one is known as a *reconfiguration*. Three steps have to take place for a successful reconfiguration:

1. Determine a set of possible system blueprints that can be employed for reconfiguration on failure by defining a set of mapping rules to determine a set of blueprints that can be used when a set of particular component failures occur.
2. Determine the events that will trigger the need for a reconfiguration and select the appropriate new system blueprint when each trigger occurs
3. Employ a mechanism by which the system transfers from the old to the new system blueprint safely.

System reconfiguration can take many forms. For instance, it may be that a smart sensor design may be able to undertake a reconfiguration activity to mask, or provide graceful degradation, of the sensing services it provides when a particular class of internal failure occurs. This type of reconfiguration comes under the heading of “adaptive embedded systems or reconfiguring embedded systems” (Trapp and Schumann, 2002). Secondly, it may be that a processor or communication bus failure is reported thus requiring functionality to be reallocated to other computing modules during run-time. In other words software functionality is moved from one computing module to another. Furthermore, it may be necessary to reduce the amount of low safety criticality functionality to allow the critical functionality to be preserved. IMS is much more flexible than existing federated systems in this respect and this type of reconfiguration is emphasised in this paper.

Ultimately, in IMS the intention is for a platform to be capable of reconfiguring its system blueprint whilst operational. The simplest approach to this is to produce

a set of system blueprints and have them available in the form of look-up tables to be used if a given trigger events occur (Nicholson, Hollow and McDermid 2000). One possible mechanism for transferring from one configuration to another is to employ an intermediate mapping that only has the processes in the old mapping that remain in the new mapping in the intermediate mapping and then to add in the new / changed processes to form the new configuration. There is a significant number of safety related issues that still need to be addressed (Jolliffe 2004) with this approach, this is an area of potential future research, but let us assume for the purposes of this paper that a set of system blueprints can be determined and that a reconfiguration mechanism can be developed.

In the discussion above it is implicit that reconfiguration can take place at a number of different levels in the system. The concept of a hierarchical reconfiguration system based on local and global reconfiguration mechanisms is relevant. The impact of a reconfiguration can therefore be very localised or extensive dependent on the nature of the event that triggers the reconfiguration. The safety argument for the reconfiguration system employed will need to be based on the properties of the reconfiguration system at the overall platform level.

The remainder of this paper focuses on step two, that is identifying a failure via health monitoring and triggering the appropriate reconfiguration process. The failure that has occurred must be identified by the appropriate part of the IMS. Recovery from this failure may involve changes to multiple elements of the IMS. The need to correctly identify faults, and which system element should be responsible for doing so, is therefore paramount to the overall effectiveness and accompanying safety argument for reconfiguration on failure. An extension to the concept of HM offers the best chance of providing the trigger for reconfiguration on failure in an IMS at an appropriate level of safety integrity.

### **3. Health monitoring**

#### **3.1 Current Practices**

Health Monitoring (HM) is a broad term used to mean a wide range of maintenance related activities including condition-based maintenance, condition monitoring, fault management and life usage monitoring of electro-mechanical components in a system. It is therefore essentially an extension to system failure diagnostics. A vast amount of work has been directed towards developing reliable and state of the art HM related techniques (Kacprzyński and Hess 2002). However, explicit HM systems have only been applied in a limited manner, for example to helicopter systems. The HM systems in helicopters, also known as Health and Usage Monitoring System (HUMS), are intended to monitor the health of helicopter rotor and rotor drive system components which are primarily concerned with mechanical parameters. They are also intended for off-line data processing for maintenance purposes.

Tanner and Crawford (Tanner and Crawford 2003) have developed an Integrated Engine Health Monitoring System for Rolls-Royce gas turbine aero-engines. They claim that as long as no other component is dispatch-critical the system brings at least two benefits:

- Reduction of operational n-service disruption by the avoidance of “surprise” failures
- Improved maintenance process resulting from greater in-service knowledge and failure identification, leading to a more selective and cost effective engine strip procedure and piece part replacement

The majority of this work has been undertaken in the aerospace domain. However the automotive industry are also interested in diagnostics for drive by wire systems. In (You and Jalics 2004) a generalised diagnostics component (GDC) and a modular hierarchical fault diagnostic strategy is developed. One advantage of the approach they recommend is that

“The architecture can be quickly and gracefully updated to a new By-wire diagnostic model on the reconfiguration of GDC’s to adapt to the changed environment.”

This reduces the overhead associated with updating the diagnostic system when the system is changed. The overhead associated with the HM system and the effort required to change and recertify the HM system when a change is made to the system will have a major impact on the viability of the reconfiguration approach to failure management. This is considered further in the future work section of this paper.

If reconfiguration is to be triggered by the results of a HM system then authority must be given to some system element to initiate the reconfiguration. This element may, or may not, be the HM system. Four levels of authority for a HM system can be envisaged (see Table 1). The current generation of HM systems can be characterised as level 3 or level 4 systems. For dynamic reconfiguration on failure to occur at run-time, that is while the system is operational, level 1 or level 2 HM is required. If reconfiguration is to take place when the system is not operational then approaches using any of the different levels of HM can be envisaged. However, if level 3 or level 4 HM systems are employed support personnel or another system would be required to make the decision to reconfigure, or not, based on the data from the HM system.

### **3.2 Health monitoring for IMS**

Computer based systems such as IMS pose particular problems for HM and the concept of reconfiguration on failure. The sensors and actuators attached to the IMS are amenable to the current generation of HM systems. However, for reconfiguration on failure a high level of authority to initiate reconfiguration would be required, which is beyond the current state of the art.

The hardware employed within an IMS such as processors and communication buses may be amenable to current HM techniques if failure is gradual, indicated by parameter deviations. However, electronic components are often subject to instantaneous failures, which poses more of a challenge for HM systems designers. Thus an extension to current HM systems is required to enhance the capabilities of

HM systems so that they can be applied to identifying failures in IMS computer based hardware. Furthermore, hardware failures in the IMS may impact on the ability of the HM system to identify failures and to reconfigure the system as the IMS also hosts the HM system. Suitable partitioning of the HM system will be required to overcome this issue.

1. Full authority	HM carries out health monitoring such as fault detection and fault analysis (i.e. extent of failure) and reports any failure or degradation in operation. It has authority to shut down equipment, applications, etc and to initiate software / hardware reconfiguration when required.
2. Semi authority	HM carries out health monitoring such as fault detection and fault analysis in operation but has no authority to shut down equipment, applications, etc. It reports any failure or degradation and recommends actions to be performed, by the operators, such as shutting down failed components, requesting immediate maintenance actions at destination or software / hardware reconfiguration.
3. Maintenance	HM carries out health monitoring such as fault detection and fault analysis and reports any failure or degradation in operation for maintenance purposes only.
4. No authority	HM carries out health monitoring such as fault detection for later analysis.

**Table 1: Levels of Health Monitoring**

IMS employs a layered approach with an operating system and separated application software. These layers are connected via an interface layer, such as that provided by ARINC 653 (ARINC 2003). It is therefore necessary to identify the types of failure that can occur in the software logic of the IMS and to identify such failures at appropriate levels in the HM system.

Failures in a component, such as a communication bus failing silent, are propagated through the IMS and may be transformed by it into other forms of failure, or may indeed be masked by the system. This propagation and transformation (Lisagor et al 2004) of a failure needs to be studied by the system designer to determine the appropriate level to place the authority to decide to initiate a reconfiguration on failure. Furthermore, the nature of this propagation and transformation may determine the form of the reconfiguration required. A technique to consider the safety implications of reconfiguration is presented in Section 4.

Thus a number of questions remain to be answered with respect to the use of HM in an IMS. Does a computer system failure such as computing component

burnt-out show gradual deterioration? Are there correlations between component failures and system crashes? If so, what are they? Can failures during execution be correlated to failures in a hardware component or flaws in software logic? The first steps towards answering these issues are presented in this section.

For HM of an IMS a distinction should be made between deviation due to malfunction of computer hardware / software and deviation due to malfunction of electro-mechanical components, such as sensors. HM for failures within an IMS should be designed following the principles adopted for IMS, which are open system principles conforming to an Application Programming Interface (API) standard. Thus, IMS health monitoring could be the function responsible for monitoring the system to detect, and report hardware (sensors, actuators, buses, processors, etc), software application and operating system faults and failures. The complexity of IMS will require a high degree of capability and authority to be invested in the HM system if reconfiguration on failure is to take place.

Standards such as ARINC 653 discuss HM as an integrated part of an IMS operating system based on various levels determined by where a fault / failure arises. ARINC 653 does not explicitly address reconfiguration on failure issues. It does however provide a good starting point. It classifies errors within an IMS according to the location of their causes: module level errors, partition / application level errors and process level errors. Module level errors include module configuration errors during initialisation, errors during partition switching and power failure. Partition level errors encompass partition configuration error during initialisation, error during process management, error during an error handler process. Examples of process level errors are memory violation and illegal operating system request. From this classification ARINC 653 defines three levels of health monitoring:

- Module level health monitoring (MHM)
- Partition / application level health monitoring (PHM)
- Global level health monitoring (GHM).

MHM provides means for detection of hardware errors in a hardware module, which are non-function related errors such as violation of partition boundary and timing overrun. PHM supports detection of specific functional application errors as well as some external hardware failures such as monitoring data from sensor. GHM provides error logging from the other HM levels and passes on information to particular modules.

ARINC 653 fails to address many important issues that should be defined for a health monitoring system of an IMS, especially those that are potentially essential for safety and certification if the HM system is to be employed as part of a reconfiguration on failure approach to fault tolerance. Examples are a definition of the roles/authority in monitoring and management of failures and responsibilities of health monitoring.

HM for reconfigurable IMS will need to be either level 1, with full authority to initiate a reconfiguration, or level 2, with operator assisted authority to reconfigure the system. In fact different failure modes will require different levels of authority. Determining this authority for each potential reconfiguration becomes an activity

as part of the design of the fault tolerance aspects of the applications hosted by the IMS. The mechanism to undertake the commanded reconfiguration would be controlled by the operating system of the IMS.

For systems that employ reconfiguration on failure controlled by the operating system in response to a trigger event four levels of health monitoring would seem to be more appropriate than the three proposed in ARINC 653. These four levels are:

- Software process / hardware component monitoring (CHM)
- Module / application level health monitoring (MHM)
- Partition /application level health monitoring (PHM)
- Global level health monitoring (GHM).

There is a strong need to define clearly the scope of each level of the HM system for reconfigurable IMS. For instance, what failures are to be identified and addressed by reconfiguration? This is known as the reconfiguration coverage of the system. An unclear definition will lead to great difficulty in certifying the IMS because reconfiguration coverage is one of the most crucial aspects for the performance of an IMS under failure. This definition of responsibility should be incorporated into a regulatory standard to assure compliance. The following scope in monitoring a reconfigurable IMS are proposed based on where a failure can arise:

- **CHM:** responsible for monitoring the health at software process / component level such as the presence of a violation, deviation of particular parameters (data from a sensor). At the component level, HM reports the status of that component. This includes deviation of values, no values, persistency of faults, and response. This is similar to built-in-test. At the software process level, HM reports violation / exception.
- **PHM:** covers violations at the partition / application level, deviations in function performance, etc.
- **MHM:** is responsible for monitoring communication, data flow between modules, etc.
- **GHM:** monitors performance related parameters, performance trends, etc.

There is a clear link between the level of authority and the scope of a HM. For instance, it may be reasonable to give full authority to a CHM to shut down a faulty piece of hardware or software process given certain failure modes of the system. It is unlikely that a CHM will also be given the authority to order a reconfiguration of the system. Reconfiguration on failure is likely to be a function of MHM or GHM depending on the architecture of the IMS. There is also a link between the propagation and transformation path of a failure and the level of authority of a particular HM to initiate reconfiguration. Work is continuing into guiding the ability to decide on the scope and authority of each level of HM within an IMS.

In this section the issues surrounding the use of HM as part of a reconfiguration on failure approach to fault tolerance for IMS have been introduced. These issues can be summarised as which failures should be identified by the HM system for reconfiguration and where in the HM system hierarchy should the decision to

undertake reconfiguration in response to a given failure event reside. In section 4 the safety and certification issues of a reconfiguration on failure approach are addressed.

## **4. Safety and Certification of HM for Reconfigurable IMS**

### **4.1 Safety aspects of HM for Reconfigurable IMS**

The ability to reconfigure on failure clearly has safety implications. If the health monitor system does not identify a failure that should be addressed via a reconfiguration activity then the system may be put into a potentially hazardous state. Alternatively, if the health monitor incorrectly initiates a reconfiguration this may also have safety implications. A safety argument will need to be developed for the HM system used as part of the deployment of the reconfiguration on failure approach. This is beyond the scope of this paper but the framework of such an argument can be found in (Jolliffe 2004).

Two safety analyses are required to assess the contribution of the HM system to a reconfiguration approach to fault tolerance for a safety critical system. The first relates to which failures should be addressed using a reconfiguration approach, where each of these failure should be addressed in the health monitoring system and where the decision to decide that a particular reconfiguration action should take place in response to the failure should be determined. In other words how can the requirements be elicited for the HM aspects of a reconfiguration on failure system? The second analysis is required to determine how failures of the HM system can contribute to failures of the reconfiguration on failure functionality as a whole. In fact the same technique can be employed to undertake both of these analyses. The chosen technique is SHARD (McDermid et al 1995), which analyses flows, such as data flows, through a system. The aim is to analyse the effects of failures that occur for a number of classes of failure namely omission, commission, early, late and value failures.

Suppose that a system is to be analysed to determine whether reconfiguration should take place as a result of a data communication bus producing a stream of incorrect messages. If the correct messages on that bus do not arrive because of the “chattering” of the bus then this is characterised as an omission failure in the SHARD technique. The extra messages may also be defined as a commission failure because they represent messages that were not required. The effects of this failure in the system context can be addressed using the SHARD technique. It may be for instance that the bus can delete all the messages thus removing the commission failure. The failure has been transformed into an omission failure. The bus may not be able to do this and the messages will then be propagated to other elements of the system. It might be for instance that in the context of the system

such extra messages can only be identified at the application level in which case this would be the appropriate level to trigger a reconfiguration.

The SHARD analysis allows the system designer to track the propagation and transformation of failures through the IMS to the end level effects. Decisions as to which failures need reconfiguration, where the failure should be identified and which element of the system should have responsibility for initiating the reconfiguration can then be determined. In this way the reconfiguration on failure approach becomes an integral part of system development. In our chattering bus example the appropriate response may be to reconfigure the system so that a redundant bus sends the messages. The sending and receiving processes would need to be able to switch from the existing bus to the new bus and timing / performance requirements would still need to be met.

Take a different example. Suppose that the system design had a smart sensor that could undertake fault diagnosis on a number of its failure modes. It is able to shut itself down or in some circumstances reconfigure itself to overcome the failure. It is able to send a message that it has done so to the IMS HM system. In this case the designer must decide whether the sensor should be given the authority to undertake shut down or reconfiguration locally. The designer will also need to consider the effect of an omission failure using the SHARD approach. The designer may also wish to consider a simple sensor and placing the failure identification and reconfiguration authority within the IMS. The analysis needs to be flexible enough to consider alternative design solutions.

There is another aspect that needs to be considered. So far our analysis has concentrated on failures in the hardware and functionality of the system, what about failures in the HM system? SHARD can help here to. Suppose that it is determined that a reconfiguration is required for the chattering bus example. Now suppose that there is a failure in the HM system and in some circumstances it erroneously identifies that the bus is in the chattering failure mode and initiates the reconfiguration. What is the effect of this at the system level? Can the system be put in a hazardous state as a result of this failure? Can another part of the HM system identify this failure? A SHARD analysis can be undertaken to determine the potential effects of such a failure mode. As another example, suppose that there is a failure mode inside the smart sensor that erroneously initiates a shut down of the sensor. Is this a safety issue or merely an availability issue?

An alternative approach, that in reality may be complementary to the SHARD analysis, is to undertake an initial system FMECA at the system design stage. (Kacprzyński and Hess 2002) argue that FMECA is a perfect link between the critical overall system failure modes and the health management system designed to help mitigate those failure modes. However, this approach does not address health management technologies for diagnosing faults and typically focuses on subsystems independently.

Current work is focusing on producing examples of SHARD analysis for a variety of hardware and software failures in a case study. It is hoped that results will be available by the conference to show how SHARD analysis can aid the designer in determining an appropriate reconfiguration on failure approach for a given system and to analyse the potential effects of failures in the proposed reconfiguration approach with respect to health monitoring.

## 4.2 Certification of HM for reconfigurable IMS

The system level at which a HM is employed and the type of failures addressed by reconfiguration partly determines the amount of effort required to certify the reconfiguration system. A full authority HM would require a significant effort to certify, as it is potentially a single point of failure. Commensurate validation and verification activities need to be determined.

The correlation between software criticality levels of the applications hosted on the IMS and HM depends on the level of authority that will be implemented for HM. There is thus a correlation between HM authority levels with software criticality level as per DO178B (RTCA/DO 1992) of the functionality being monitored. For example, if HM has full authority to initiate a reconfiguration of a system with software of criticality level A then the HM system must also be at criticality level A. Table 2 lists the likely correlation between HM authority levels and software criticality levels. It suggests the appropriate HM criticality levels.

Authority	Software criticality level of system				
	Catastrophic	Hazardous	Major	Minor	Non-essential
Full	A	A/B	C	D	E
Semi	A/B	B/C	C/D	D	E
Maintenance	C	C	D	D	E
No	D	D	D	D	E

**Table 2: HM authority Versus Criticality**

So in this section the requirement to consider the safety effects of a decision to employ a reconfiguration on failure approach to fault tolerance on a particular IMS has been investigated. First, it is proposed that a design time SHARD analysis should be undertaken to identify the failures that will trigger a reconfiguration and the appropriate level in the HM system hierarchy to place the reconfiguration decision. Secondly, a SHARD analysis should also be undertaken to assess the safety implications of a failure in the HM system. Finally, validation and verification activities commensurate with the authority of the HM and the safety criticality of the applications it can affect should be undertaken.

## 5. Conclusions

This paper has considered the first steps towards implementing a reconfiguration on failure-based approach to fault management in an IMS. It has concentrated on the scope, authority and safety / reliability analysis of a HM system required to initiate reconfiguration on failure. A four level HM is proposed with some elements being given full authority to initiate a reconfiguration and others only able to implement local actions. Safety analysis is via two SHARD analyses. The

integrity level that would need to be associated with each level of HM depending on the authority of each level has also been addressed.

A significant number of unresolved issues have emerged as a result of this work that must be addressed before this approach can be used in real systems. For example, will the GHM have the final responsibility for initiating a reconfiguration based on the actions of other HM or can lower level HMs do this? How can the HM be improved to predict failures and what sort of reconfiguration would be appropriate in these conditions? How can the HM be removed as a potential single point of failure to a hazardous system state? How will the reconfiguration approach be certified?

Furthermore, what is the impact of a change to the system on the HM regime? Once a system is in operation it will be subject to change throughout its lifetime. Indeed one of the perceived advantages of IMS is the ability to implement incremental change. The aim here is for a change to have a minimal, and well defined, impact on elements of the IMS not directly affected. Work is needed to determine how the HM system can respect this approach as much as possible. This will imply that the impact on the HM system should be minimal if a change is made and that the amount of reworking of the safety analysis should be as proportionate to the size of the change as possible.

These issues present a significant challenge to the reconfiguration on failure approach put forward in this paper and may dilute the gains in reliability / safety that can be achieved as a result of the graceful degradation ability provided by reconfiguration on failure approach. Nevertheless, there is a strong incentive to resolve these problems.

## References

- ARINC (2003). ARINC SPECIFICATION 653-1 Avionics Application Software Standard Interface, 2003
- ASAAC (2002). ASAAC Phase II Stage 2, Second draft of proposed guidelines for system issues - Volume 2: Fault Management, REF-WP: 32350, 2002
- Bates SA, Bate IJ, Hawkins RD, Kelly TP, McDermid JA (2003). Safety Case Architectures to Complement a Contract-Based Approach to Designing Safe Systems. In Proceedings of 21st International System Safety Conference, Ottawa, Canada, 2003.
- Eurocae (2004). Modular Avionics Guidance Document revision F. Ref: EUROCAE WG60 / RTCA SC 200, Aug 2004
- Globalsecurity (2004). F-22 Raptor Flight Control Systems. [www.globalsecurity.org/military/systems/aircraft/f-22/fcas.htm](http://www.globalsecurity.org/military/systems/aircraft/f-22/fcas.htm), 7<sup>th</sup> October 2004
- Jolliffe G (2004). "Exploring the Possibilities of Safety Case Production for IMA Blueprints. MSc Project, Department of Computer Science, University of York, 2004
- Kacprzyński GJ and Hess AJ (2002). Health Management System Design: Development, Simulation and Cost/Benefit Optimization. IEEE, Big Sky, MT, March 2002

- Lisagor O, Pumfrey DJ & McDermid JA (2004). Safety Analysis of Software Architectures - "Lightweight PSSA", Proceedings of the 22nd International System Safety Conference, Providence, RI, System Safety Society, P.O.Box 70, Unionville, VA 22567-0070, USA, 2004
- McDermid JA, Nicholson M, Pumfrey DJ & Fenelon P. (1995). Experience with the application of HAZOP to computer-based systems, COMPASS '95: Proceedings of the Tenth Annual Conference on Computer Assurance, Gaithersburg, MD, pp. 37-48, IEEE, ISBN 0-7803-2680-2, 1995
- Nicholson M, Hollow P and McDermid JA (2000). Approaches to Certification of Reconfigurable IMA Systems. INCOSE 2000, Minneapolis, USA, July 2000
- RTCA/DO (1992). RTCA/DO-178B - Software Considerations in Airborne Systems and Equipment Certification. RTCA/DO, 1992
- Tanner and Crawford (2003). Aircraft Airborne Condition Monitoring. British Energy. IEE meeting, Gloucester, United Kingdom, 14 May 2003
- Trapp M and Schurmann B (2002). On the Modelling of Adaptive Systems. Sigsoft 2002/FSE10, Charleston, SC, USA, 2002
- You S and Jalics L (2004). Hierarchical Component Based Fault Diagnostics for By-Wire Systems. SAE white paper, SAE, 2004-01-0285, 2004
- VICTORIA (2001). VICTORIA project on the Europa website. [europa.eu.int/comm/research/growth/gcc/projects/in-action-victoria.html](http://europa.eu.int/comm/research/growth/gcc/projects/in-action-victoria.html), 2001