The Maintenance Aware Design environment: Development of an Aerospace PHM Software Tool

Andrew Hess, Jacek S. Stecki and Shoshanna D. Rudov-Clark

***Abstract*—The Maintenance Aware Design environment (MADe) was conceived to provide a suite of software tools that could be used to design, assess and optimise Prognostics and Health Management systems for use in a wide variety of high risk industries where safety and reliability are critical, including mining, offshore and aerospace applications. MADe is currently being developed for application to aerospace systems and beta testing of the software tools is due to commence in mid-2008. This paper presents the concepts underpinning the system modelling, failure database generation and monitoring design tools using case studies of subsystems and components relevant to aerospace applications.**

Index Terms — **Prognostics and Health Management, Software, FMECA database**

I. Introduction

Aircraft and avionics, offshore, marine and other complex engineering systems often operate in harsh environmental and operational conditions and, especially in the military field, must meet stringent requirements of reliability, safety and maintainability. With maintainability now being a major constraint in the development of new systems it is important to develop techniques to monitor the health of the system, to diagnose system problems prior to failure and predict the system's remaining life. These techniques are encompassed in the Machine Condition Monitoring (MCM) and/or Condition Based Maintenance (CBM) approaches to system maintenance. The last twenty years has seen great progress in the development of new sensing techniques, diagnostic and prognostic methodologies and in the application of computer analysis techniques. Nevertheless it is interesting to note that, at a 2002 Workshop on Condition Based Maintenance organised by The Advanced Technology Program of the National Institute of Standards and Technology (NIST) USA, the following barriers to widespread application CBM were identified:

- Inability to accurately and reliably predict the remaining useful life of a machine.
- Inability to continually monitor a machine.
- Inability of maintenance systems to learn and identify impending failures and recommend what action should be taken.

These barriers can be redefined as deficiencies in prognostics, sensing and reasoning. These and other limitations of current implementations of CBM were, of course, recognised by others and led to the development of programs (e.g. in the military) aimed at overcoming them.

Thus CBM evolved into a new concept - Prognostics and Health Management (PHM) which has two basic elements:

- Prognostics - predictive diagnostics, which includes determining the remaining life or time span of 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.

PHM must be addressed in the context of system design, maintenance approaches, failure analysis, model based monitoring and artificial intelligence technology. Fusion of these techniques and technologies with design processes will lead to improved reliability and maintainability of systems.

The following are key characteristics of Prognostics and Health Management according to Scheuren et al [1]:

- Shift from time-based to on-condition maintenance through the use of advanced condition monitoring techniques.
- Identification of probable/possible failures or problems in systems and their criticality using FMEA, FMECA and other

techniques in early design stages of new systems, auditing of existing (legacy) systems.
- Maximising lead-time for unscheduled maintenance via the application of advanced diagnostics and prognostics systems.
- Real-time information transfer about upcoming maintenance events.
- Employing an open architecture for PHM systems to easily incorporate new technologies (e.g.: sensors, methods and techniques).

The design philosophy of MADe recognises the need for reducing Total Ownership Costs (TOC) or Life Cycle Costs (LCC) of systems through improved system reliability and maintenance planning via:

- Integrating PHM into system design to avoid inherent system functional deficiencies and reduce the duration (and cost) of the design cycle.
- Knowledge Transfer through reusable, scaleable system models and enhanced collaboration within a distributed engineering environment.
- Identifying and developing appropriate systems to prevent critical potential failure modes.
- Utilisation of 'Best of Breed' technologies such as genetic algorithms and neural nets to provide significantly faster system analysis.
- Enabling planned maintenance operations based on residual life estimations from real-time sensors based on the results of live system diagnostics.

The MADe system capabilities are shown in figure 1. The system modelling tool (Failure Knowledgebase or the FMECA Database Generation Tool) generates hardware and functional system models which can be used to predict the system response to component level faults and process their criticality. The tool aims to provide a rapid and affordable means of generating and continually updating system and failure knowledge bases.

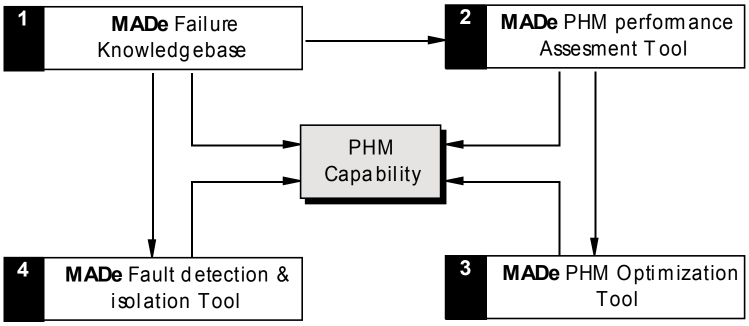

1
MADe Failure Knowledgebase
2
MADe PHM performance Assesment Tool
PHM Capability
4
MADe Fault detection & isolation Tool
3
MADe PHM Optimization Tool

Fig. 1. MADe software capabilities

The system monitoring design tools (PHM Performance Assessment & PHM Design Optimization) are used to optimise the number and location of sensors in energy transmission systems and to enable real-time 'what-if' analysis to determine the impact of trade-offs, such as weight reduction via reducing the number of sensors, on the fault coverage. The Advanced Fault Detection & Isolation Tool (Model Based Diagnostic [MBD]) uses the hardware system model and system monitoring design tools to generate an MBD application. The application uses artificial intelligence and model-based simulations of the system to offer advanced real-time diagnostic analysis.

This paper concentrates on the development of points 1 and 3 in figure 1, which is the generation of failure knowledge bases (1. MADe FMECA Database Generation Tool) and the optimization of system monitoring designs (3. MADe PHM Design Optimisation Tool).

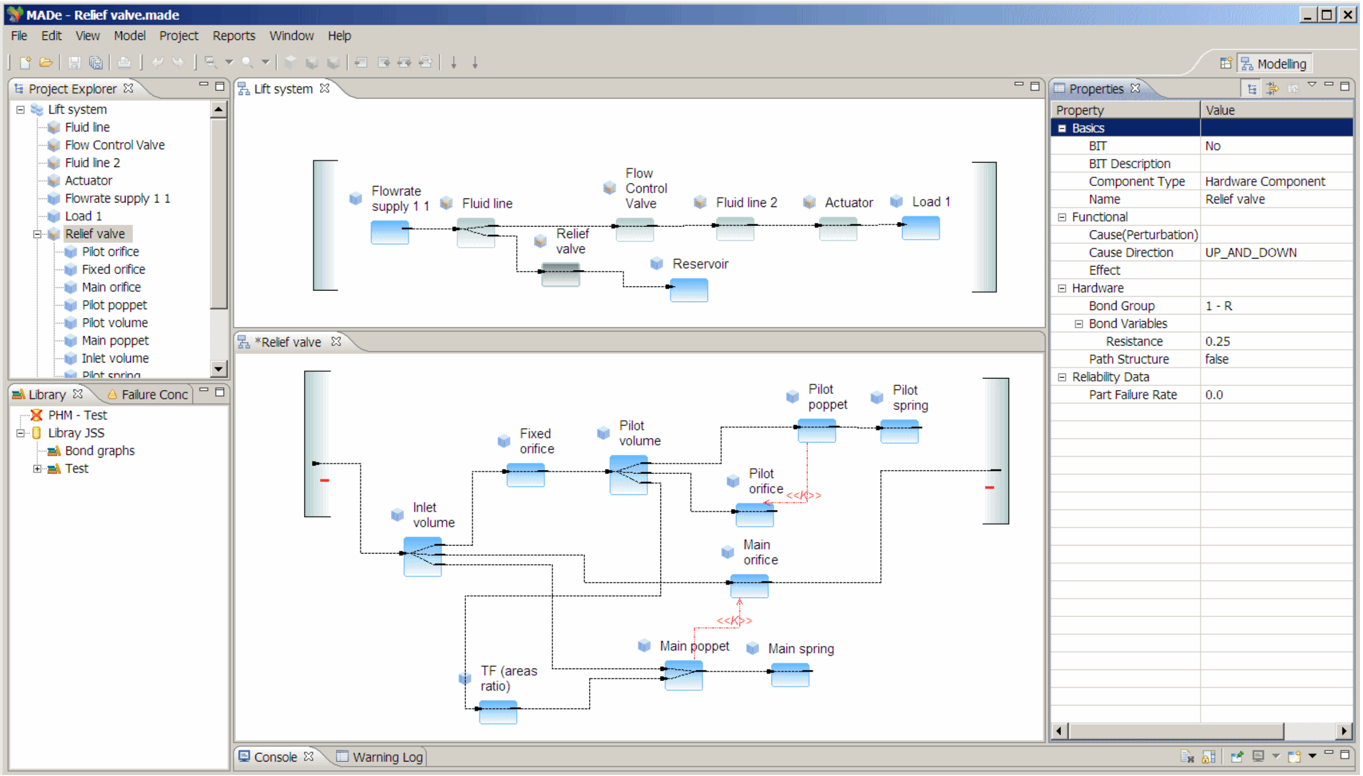
II. Modelling and failure knowledge-bases

The availability of failure knowledge-bases is a basic requirement for generating PHM systems capable of fulfilling their objectives. This includes a full knowledge of failures which can be generated during system operation. These failures may arise due to defects within a component or from the effects of other component failures that propagate through the system.

This information must be available before attempts to design model-based diagnostic and prognostic reasoners are made. Incomplete knowledge of failures will result in reduced reliability of such diagnostics/prognostics reasoners. This knowledge is also crucial to the design of sensing systems. If the sensors are designed to cover an incomplete set of symptoms then the diagnostic and prognostic capability of any PHM system is degraded.

The MADE system modelling tool and failure knowledge-base aims to identify potential operational and diagnostic problems in the conceptual stages of system design and provide a guide to make the necessary capability and requirements trade-offs to optimise the design of the final system. System models can be generated at the conceptual design phase using functional descriptions of the components and subsystems. Mature designs can be input using a CAD interface which allows the user to export system information to MADe and import the analysis results to view within the CAD environment. By including component hardware details and defining the operating modes, the model can be used to determine system monitoring requirements and to formulate Model Based Diagnostic applications for real time system Fault Detection and Isolation (FDI).

Figure 2 provides a screenshot of the MADe user interface for system modelling and failure knowledge base generation. An hydraulic lifter has been modeled using hardware components. The central panes show the assembly of the system model, in which each level of indenture can be opened and modelled in detail. For example, the relief valve is highlighted in the system model (upper central pane) and has been opened to view its sub-model in lower central pane. Each component also possesses a mapping of its internal failures which define the physical process by which each failure cause evolved to a component failure mode. This provides a causal mapping between inputs and disturbances to the component and the associated failure modes. The right hand side panes show failure concept maps for the side plate and needle bearings. At the bottom of the window criticality data is displayed for the needle bearings.



Property	Value
Basics	
BIT	No
BIT Description	
Component Type	Hardware Component
Name	Relief valve
Functional	
Cause(Perturbation)	
Cause Direction	UP_AND_DOWN
Effect	
Hardware	
Bond Group	1 - R
Bond Variables	
Resistance	0.25
Path Structure	false
Reliability Data	
Part Failure Rate	0.0

Fig. 2. MADe system modelling interface

A. Component Modelling

A MADe system component is defined by its name and synonyms, a functional description and hardware details such as bond graph type (resistor, capacitor etc.) and dynamic coefficients. The component operates within a subsystem or system via its interfaces to adjoining components and the operating environment. The attributes of a MADe library component include:

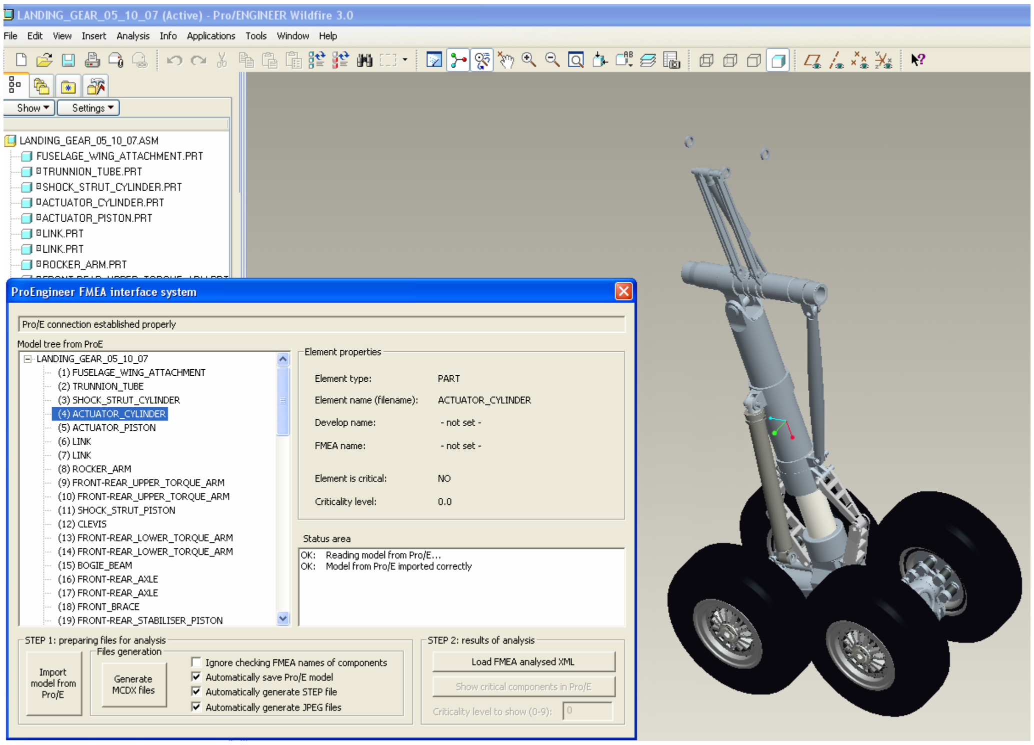
- Name, synonyms, location within subsystems/systems
- Description of its function and flows
- Hardware details

Hardware component details define the component in terms of the generation, transmission or conversion of energy. The hardware component description contains data relating to its interfaces (inports and outports), port-mapping and dynamic properties. Controllable attributes, such as energy flows and operating modes, and uncontrollable parameters such as temperature and viscosity are stored in the component database. Library components possess a Failure Concept Map which represents the sequence of failure events from the root causes to component-level failure.

B. CAD Interface

A CAD interface has been developed for CATIA, Solid Works and Pro Engineer. This enables MADe to automatically accept system component data and revisions to the system design from engineering design software. Engineers can generate a MADe subsystem model by exporting subsystem data from CAD and using the information to construct a system block diagram. The results of a system failure analysis can be exported back to the CAD environment and viewed on the original subsystem drawing.

The CAD Interface prepares a CAD subsystem drawing for export by listing the components in a hierarchical tree that indicates the hierarchical level of each component within the system, as shown in figure 3. The Interface matches each component name with MADe component terminology from an imported list of MADe component names and synonyms. If a matching MADe name cannot be found, a generic placeholder is assigned to prompt the user to create a User-Defined Component in MADe.


LANDING_GEAR_05_10_07 (Active) - Pro/ENGINEER Wildfire 3.0
File Edit View Insert Analysis Info Applications Tools Window Help
Show
Settings
ProEngineer FMEA interface system
Import model from Pro/E
Generate MCDX files
Load FMEA analysed XML
Show critical components in Pro/E

LANDING_GEAR_05_10_07.ASM
FUSELAGE_WING_ATTACHMENT.PRT
TRUNNION_TUBE.PRT
SHOCK_STRUT_CYLINDER.PRT
ACTUATOR_CYLINDER.PRT
ACTUATOR_PISTON.PRT
LINK.PRT
LINK.PRT
ROCKER_ARM.PRT

Pro/E connection established properly

Model tree from ProE

LANDING_GEAR_05_10_07
(1) FUSELAGE_WING_ATTACHMENT
(2) TRUNNION_TUBE
(3) SHOCK_STRUT_CYLINDER
(4) ACTUATOR_CYLINDER
(5) ACTUATOR_PISTON
(6) LINK
(7) LINK
(8) ROCKER_ARM
(9) FRONT-REAR_UPPER_TORQUE_ARM
(10) FRONT-REAR_UPPER_TORQUE_ARM
(11) SHOCK_STRUT_PISTON
(12) CLEVIS
(13) FRONT-REAR_LOWER_TORQUE_ARM
(14) FRONT-REAR_LOWER_TORQUE_ARM
(15) BOGIE_BEAM
(16) FRONT-REAR_AXLE
(17) FRONT-REAR_AXLE
(18) FRONT_BRACE
(19) FRONT-REAR_STABILISER_PISTON

Element properties

Element type:	PART
Element name (filename):	ACTUATOR_CYLINDER
Develop name:	- not set -
FMEA name:	- not set -
Element is critical:	NO
Criticality level:	0.0

Status area

```
OK:   Reading model from Pro/E...
OK:   Model from Pro/E imported correctly
```

STEP 1: preparing files for analysis

Files generation

☐ Ignore checking FMEA names of components
☑ Automatically save Pro/E model
☑ Automatically generate STEP file
☑ Automatically generate JPEG files

STEP 2: results of analysis

Criticality level to show (0-9): 0

Fig. 3. User interface for the CAD system model import function

The MCDX, JPG and STEP export files are generated by the Interface and can then read by MADe to provide the components in the MADe system modelling workspace. The user can view jpg images of components and subsystems and can open the exported STEP file using external STEP viewing software. A hardware or functional model can then be built from the CAD components.

C. Functional System Models

Functional models are built by selecting generic MADe library components, known as functional areas, and linking them to create a block diagram. The links represent functional relationships between the components, and these functional relationships are expressed using the functional ontology developed by Stone and Wood [2]. The functional description is a two-part verb-noun statement which is formed by selecting one verb and one or more nouns from a standard list of terms. Figure 4 illustrates the selection of the function verb and the input/output flow noun to define the function of a drive shaft, which is “transmit rotational energy”.

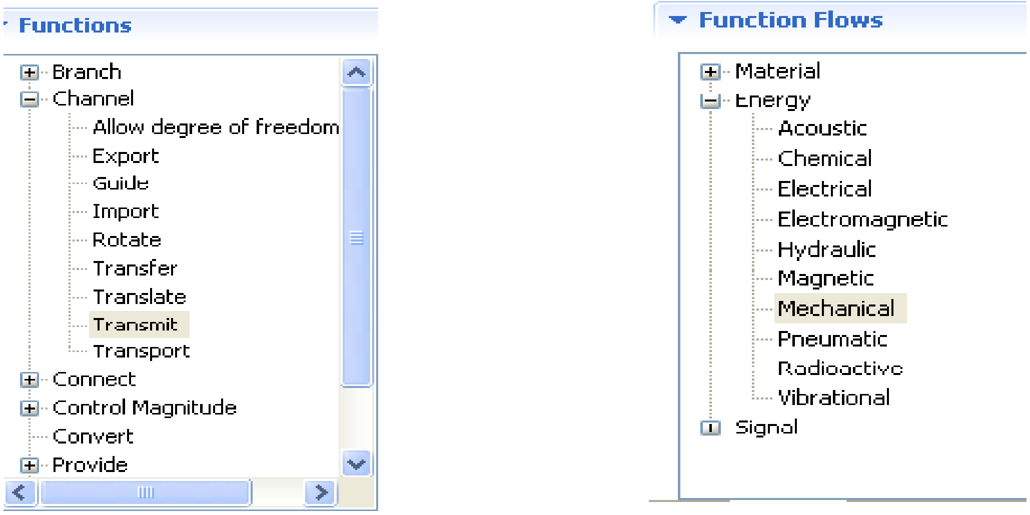


Fig. 4. Selection of functions and flows for a drive shaft

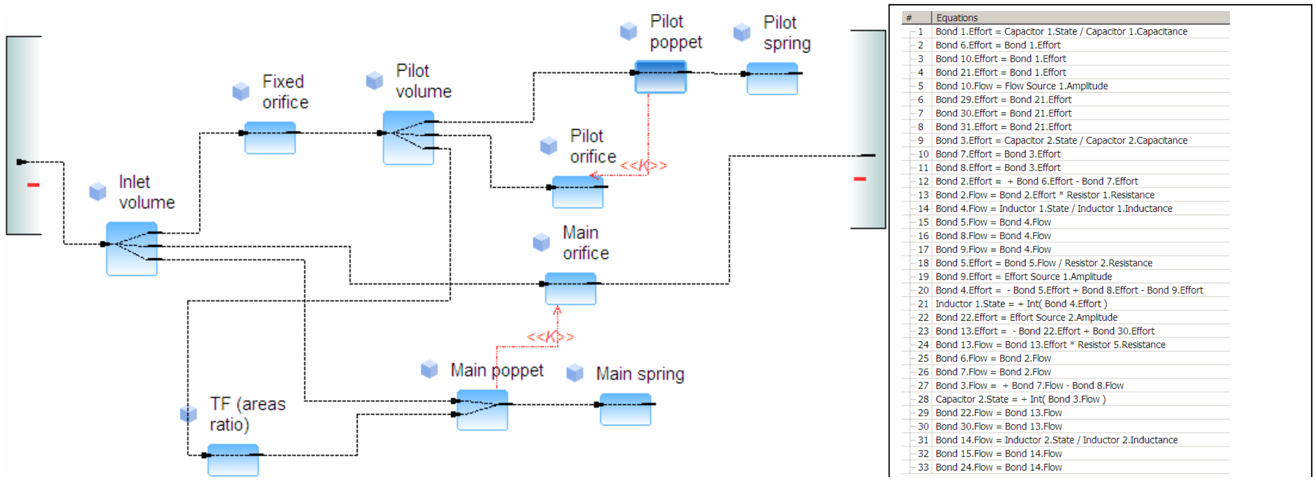
Functional areas are connected in the modelling area by lines (edges) which represent the functional flows between components. Multi-level systems can be modelled by constructing two-dimensional block diagrams within each hierarchical level and specifying the causal links between each level by mapping their inputs and outputs.

MADe automatically converts the block diagram to a directed graph, known as a concept map, which is used to propagate flows through the system. The links between functions represent the causal relations between functional components and these are given strength weightings. Functional failures describe the deviation of the behaviour of a component from its intended purpose. This is achieved by stating the changes to the output flows specified in the function. Functional failures are propagated through the system model via the system Functional Concept Map which alters the input and output flows of each component in turn according to the causal linkages between the components and the system hierarchy levels.

D. Hardware System Modelling

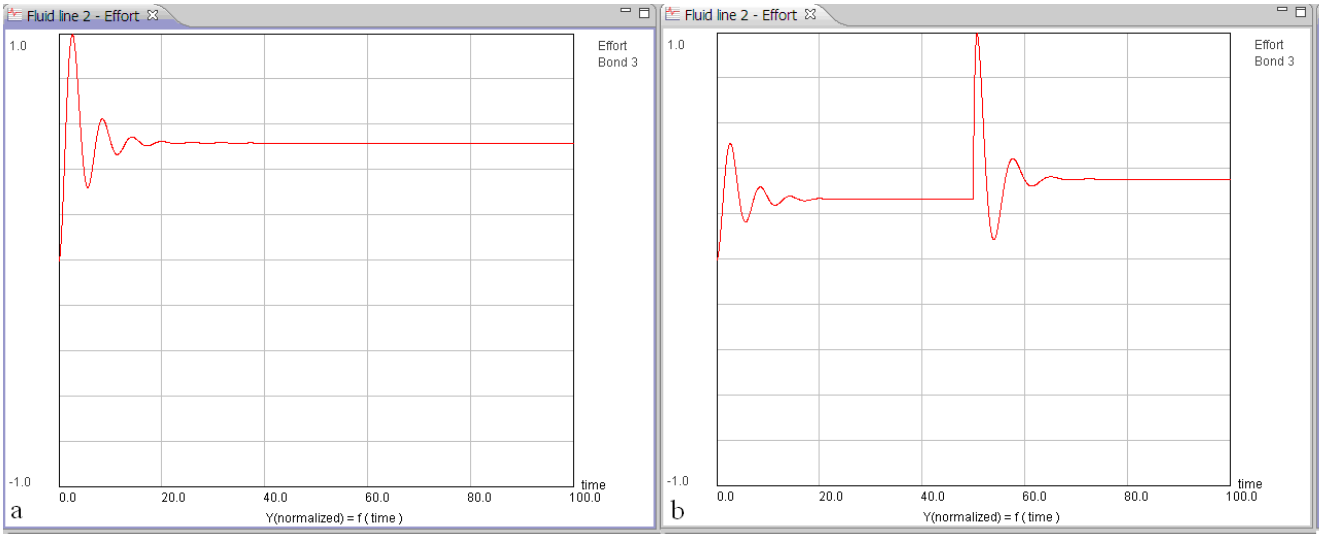
Hardware system models are constructed in MADe by selecting generic mechanical, hydraulic or electrical components from the MADe component library and adding them to the modelling canvas. The MADe library components possess default properties such as energy inputs and outputs, port-mappings, and dynamic coefficients which can be checked and adjusted by the user. A subsystem is constructed by connecting the hardware component blocks and specifying the ‘end-effect’ of the subsystem. The end-effect provides a route for transferring the output of the subsystem to the input of the next hierarchical level. MADe automatically converts hardware block diagrams into to simulation models and the causality of the model is checked to ensure it is a valid model for analysis.

During conversion the mathematical model of power flow through the system is generated as a set of state equations using the bond graph method. Figure 5 presents the simulation model of a hydraulic relief valve within the Puma landing gear system. The sub-model has been automatically converted from the hardware system block diagram to a bond graph model. The right hand pane shows the dynamic state equations that were generated for each component. The dynamic response of a component can be checked by running a test simulation. The results of a test simulation for a fluid line are shown as time response graphs in figure 6. The time response graph displays the dynamic response of the hydraulic line. At t=0 s the graph shows the transient response of the line due to system start-up, and at t= 50 s the response of the line due to failure of the flexible coupling (as an energy perturbation) is shown. This process is automatically repeated to determine the response of every component within a system hierarchy level to the failure of the flexible coupling.



#	Equations
1	Bond 1.Effort = Capacitor 1.State / Capacitor 1.Capacitance
2	Bond 6.Effort = Bond 1.Effort
3	Bond 10.Effort = Bond 1.Effort
4	Bond 21.Effort = Bond 1.Effort
5	Bond 10.Flow = Flow Source 1.Amplitude
6	Bond 29.Effort = Bond 21.Effort
7	Bond 30.Effort = Bond 21.Effort
8	Bond 31.Effort = Bond 21.Effort
9	Bond 3.Effort = Capacitor 2.State / Capacitor 2.Capacitance
10	Bond 7.Effort = Bond 3.Effort
11	Bond 8.Effort = Bond 3.Effort
12	Bond 2.Effort = + Bond 6.Effort - Bond 7.Effort
13	Bond 2.Flow = Bond 2.Effort * Resistor 1.Resistance
14	Bond 4.Flow = Inductor 1.State / Inductor 1.Inductance
15	Bond 5.Flow = Bond 4.Flow
16	Bond 8.Flow = Bond 4.Flow
17	Bond 9.Flow = Bond 4.Flow
18	Bond 5.Effort = Bond 5.Flow / Resistor 2.Resistance
19	Bond 9.Effort = Effort Source 1.Amplitude
20	Bond 4.Effort = - Bond 5.Effort + Bond 8.Effort - Bond 9.Effort
21	Inductor 1.State = + Int(Bond 4.Effort)
22	Bond 22.Effort = Effort Source 2.Amplitude
23	Bond 13.Effort = - Bond 22.Effort + Bond 30.Effort
24	Bond 13.Flow = Bond 13.Effort * Resistor 5.Resistance
25	Bond 6.Flow = Bond 2.Flow
26	Bond 7.Flow = Bond 2.Flow
27	Bond 3.Flow = + Bond 7.Flow - Bond 8.Flow
28	Capacitor 2.State = + Int(Bond 3.Flow)
29	Bond 22.Flow = Bond 13.Flow
30	Bond 30.Flow = Bond 13.Flow
31	Bond 14.Flow = Inductor 2.State / Inductor 2.Inductance
32	Bond 15.Flow = Bond 14.Flow
33	Bond 24.Flow = Bond 14.Flow

Fig. 5. Sub-model of the hydraulic relief valve with dynamic equations generated for hydraulic system



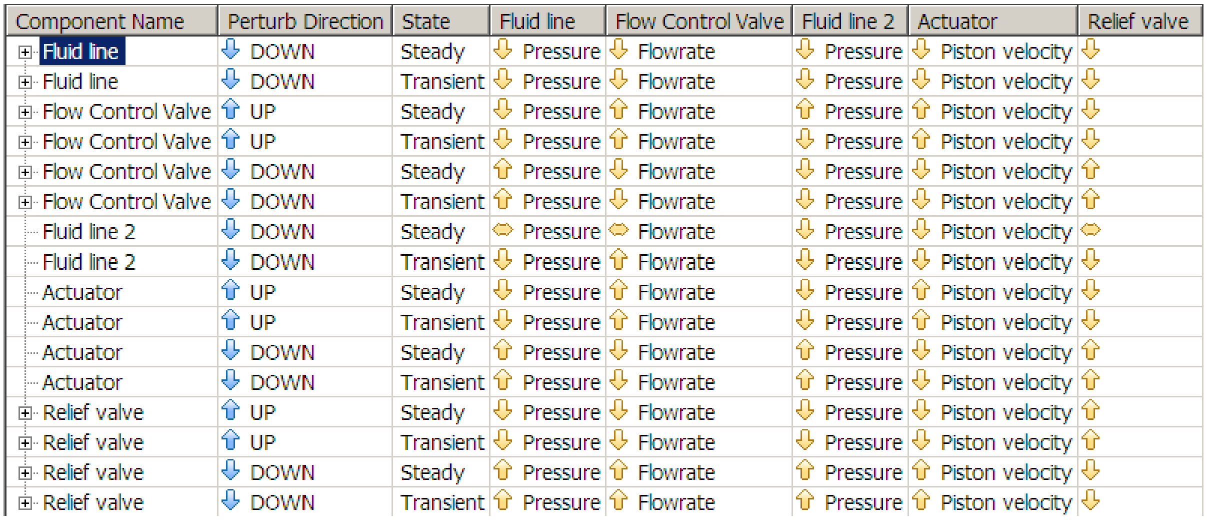
Fig, 6. Dynamic response of fluid line (a) unperturbed (b) perturbed at t=50s

Hardware failures are defined as functional changes to a component's behaviour and are represented as changes to the energy level of the component. These so-called energy perturbations are propagated through the system via the power-bonds between components and the causal links between hierarchy levels. The system effects of a component failure are identified by non-dimensional changes in energy flow levels throughout the system.

Figure 7 shows the analysis results of MADe's automatic propagation of an energy disturbance due to failure of a fluid line within the Puma landing gear system. The results are presented in table format known as the propagation analysis table, which displays the transient and steady-state response of each component as a positive, negative or zero change in energy level.

E. Criticality Analysis

The criticality ranking of system failures is used to prioritise potential component failures for redesign or remedial action. Failure criticality is assessed in the same manner for both hardware and functional models. The criticality properties are defined at the component level and propagated through the system using Failure Concept Maps (FCM). The criticality of a failure at the system level is determined by analysing the entire causal chain of events from the component level through to the system level. The established measures of criticality include the Risk Priority Number, Criticality matrix and Failure Assessment Index [3,4]. The RPN is found by combining the occurrence and detectability of a failure mode at the component level (root cause level) with the severity of its system level effect. These O, S and D values can be input as crisp numerical rankings from 1 (least critical) to 10 (most critical) or as fuzzy values, and are processed accordingly. Default values are provided by MADe and are stored as properties of a failure mode-effect pair in the MADe library.



Component Name	Perturb Direction	State	Fluid line	Flow Control Valve	Fluid line 2	Actuator	Relief valve
⊞ Fluid line	⇩ DOWN	Steady	⇩ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇩
⊞ Fluid line	⇩ DOWN	Transient	⇩ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇩
⊞ Flow Control Valve	⇧ UP	Steady	⇩ Pressure	⇧ Flowrate	⇧ Pressure	⇧ Piston velocity	⇩
⊞ Flow Control Valve	⇧ UP	Transient	⇩ Pressure	⇧ Flowrate	⇩ Pressure	⇧ Piston velocity	⇩
⊞ Flow Control Valve	⇩ DOWN	Steady	⇧ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇧
⊞ Flow Control Valve	⇩ DOWN	Transient	⇧ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇧
Fluid line 2	⇩ DOWN	Steady	⇔ Pressure	⇔ Flowrate	⇩ Pressure	⇩ Piston velocity	⇔
Fluid line 2	⇩ DOWN	Transient	⇩ Pressure	⇧ Flowrate	⇩ Pressure	⇩ Piston velocity	⇩
Actuator	⇧ UP	Steady	⇩ Pressure	⇧ Flowrate	⇩ Pressure	⇧ Piston velocity	⇩
Actuator	⇧ UP	Transient	⇩ Pressure	⇧ Flowrate	⇩ Pressure	⇧ Piston velocity	⇩
Actuator	⇩ DOWN	Steady	⇧ Pressure	⇩ Flowrate	⇧ Pressure	⇩ Piston velocity	⇧
Actuator	⇩ DOWN	Transient	⇧ Pressure	⇩ Flowrate	⇧ Pressure	⇩ Piston velocity	⇧
⊞ Relief valve	⇧ UP	Steady	⇩ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇧
⊞ Relief valve	⇧ UP	Transient	⇩ Pressure	⇩ Flowrate	⇩ Pressure	⇩ Piston velocity	⇧
⊞ Relief valve	⇩ DOWN	Steady	⇧ Pressure	⇧ Flowrate	⇧ Pressure	⇧ Piston velocity	⇩
⊞ Relief valve	⇩ DOWN	Transient	⇧ Pressure	⇧ Flowrate	⇧ Pressure	⇧ Piston velocity	⇩

Fig. 7. Propagation analysis table for hydraulic relief valve

The criticality number is calculated by combining the part failure rate for a component with the failure mode and failure effect probability [3]. Following the recommendations of the Reliability Analysis Center, the criticality number is calculated for every system level effect of a component failure mode, rather than just for the most critical effect [4]. The criticality number is therefore the property of every system level effect of a component failure mode, and provides an estimate of its likelihood. When plotted against the severity classification of the system effect, the ordinates of the data point can be used to calculate the Failure Assessment Index for the failure effect. For multi-level systems, the probability of a system-level effect is calculated by propagating the occurrence value for a component failure mode through the system using a Failure Concept Map. This map constructs the failure paths linking each mode-effect pair from the component level through the system hierarchy to the system level. Using fuzzy propagation techniques the beta value, or probability of a failure effect at system level, can be estimated. In MADe this value is referred to as the “apparent occurrence” of a failure mode, so named because it is the occurrence as viewed from the system-wide perspective.

Failure knowledge-bases contain data on library components that are either generic or component-specific. This data is associated with components represented by their icons. The results of failure analysis on components, subsystems or systems are also stored in knowledge-bases and can be used in subsequent analysis.

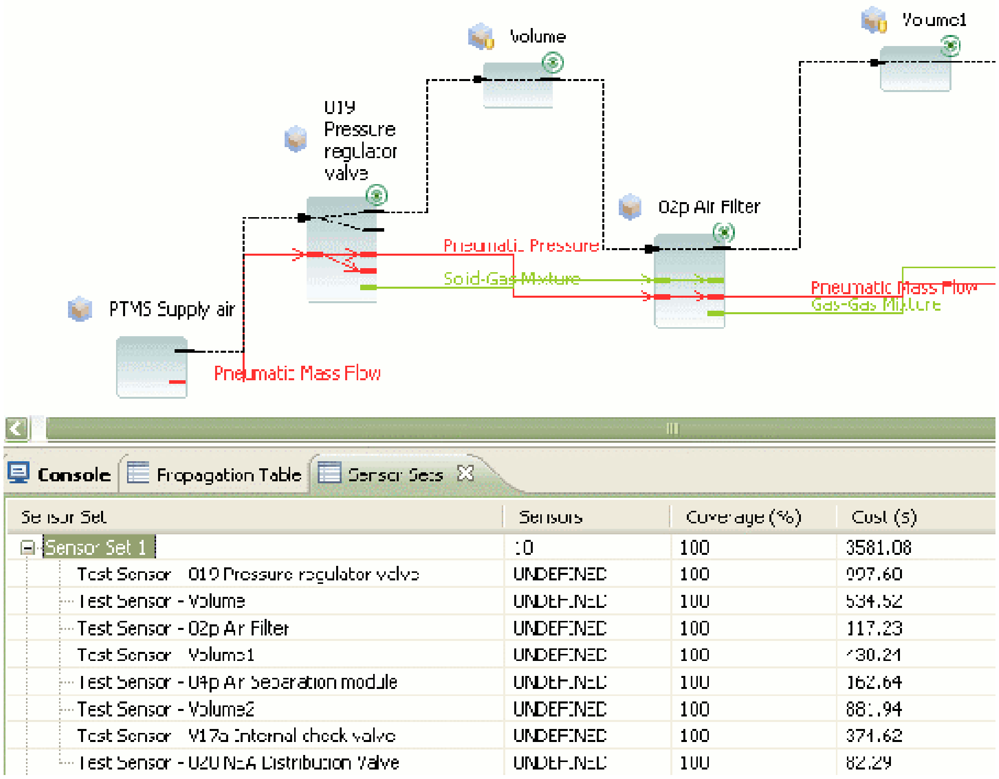
III. System monitoring

MADe provides design definition capability which enables the user to:

- Define appropriate sensor sets that fulfill stated Fault Detection and Isolation (FDI) performance specifications
- Optimise the number and location of sensors that can fulfill a specified level of FDI coverage
- Assess the FDI coverage of proposed or existing PHM system against performance specifications
- Assess the impact of changing the number or location of sensors, or the capability of sensors on the FDI coverage of the system
- Assess changes to FDI requirements due to system upgrades and design modifications

Sensor set design is accomplished using the MADe hardware system model of a system and the associated component failure database. Prior to designing the sensor set any component Built-In Testing equipment (BIT) or pre-existing sensors can be added in the model to avoid duplication. If a Model-based diagnostic application exists for the system, virtual sensors can also be included in the model by representing them as a number of ‘physical’ sensors that provide the equivalent coverage. The user can nominate the required coverage by selecting a threshold level of criticality and can also identify ‘mandatory’ and ‘excluded’ sensor locations.

Figure 8 shows a screenshot of the sensor set design and optimisation interface. The system model is displayed in the upper pane. When the design process is initiated MADe automatically generates multiple candidate sensor sets that can provide complete coverage of the system. In the lower pane the candidate sets of sensors are displayed listed along with the number of sensors in each set, total cost, and coverage. Variables such as reliability and weight can also be listed, with the purpose of being able to sort the list by these variable for optimisation. Sensor set 1 has been selected and an expanded view of this set is displayed. Each sensor is listed by the name of its location.



Sensor Set	Sensors	Coverage (%)	Cost ($)
Sensor Set 1	10	100	3581.08
Test Sensor - 019 Pressure regulator valve	UNDEFINED	100	997.60
Test Sensor - Volume	UNDEFINED	100	534.52
Test Sensor - 02p Air Filter	UNDEFINED	100	117.23
Test Sensor - Volume1	UNDEFINED	100	430.24
Test Sensor - 04p Air Separation module	UNDEFINED	100	162.64
Test Sensor - Volume2	UNDEFINED	100	881.94
Test Sensor - V17a Internal check valve	UNDEFINED	100	374.62
Test Sensor - 020 NEA Distribution Valve	UNDEFINED	100	82.29

Fig. 8. Sensor selection interface

Symptoms of a fault are all the observable energy perturbations by which a fault can be detected throughout the system, including the original failure mode of the faulty component. The diagnostic set of a fault is a list of symptoms that can be used to uniquely identify the fault. A fault/symptom table is generated that lists every previously identified fault and correlates it to its symptoms, and this table is used to generate the sensor sets. Sensor sets are generated from the fault/symptom table by optimizing (minimising) the number of symptoms (i.e.: the sensors) while preserving the observability of every fault. The observability of a fault is determined by the presence of at least one of its diagnostic sets within a given sensor set design for the system. Diagnostic rules are developed which identify the response signal values that will be detected for each fault.

The optimisation process begins at the highest level of indenture: using the MADe sensor library, and any sensors that the user has added to the library, MADe automatically generates a number of potential designs at the system level. Each level of the system hierarchy is then opened in turn for automatic optimisation/minimisation number of sensors within each level. At this stage optimisation can occur, although the sensor parameters that can be optimised depend on the level of detail in the system model. For conceptual system designs the failure symptoms are identified as system element energy outputs therefore sensors are specified in terms of the energy domain of the failure symptoms.

Sensor sets can be ranked according to the total number of sensors and the number of sensors by type (energy domain). Generalised criteria can also be applied to the energy domain categories of sensors by attributing the range of properties belonging to sensors in each of the energy domains. The criteria are developed by statistical analysis of a broad range of sensors within the MADe sensor library. For mature system designs MADe provides failure symptom details such as the specific output energy or internal loss that is being measured.

An existing system monitoring design can be modelled in MADe using a ‘drag and drop’ process in which the user selects predefined or user-defined sensors from the MADe sensor library. The performance assessment criteria are set, for example, the % of faults to be covered or confidence levels for a specified fault coverage, and the performance assessment is automatically conducted by MADe using a ‘reverse fault propagation’ technique. This method enables rapid, on-screen ‘what-if’ analysis as a 2-step process:

- edit the sensor set according to the proposed design
- re-run observability check to determine failure/fault coverage

Reports are automatically generated in which the performance criteria, details of the system model and details of the PHM sensor set are documented. Where ‘what-if’ scenarios are run, each iteration of the PHM design is documented. MADe can generate a summary report that includes the selected criticality threshold, a list of all failure modes covered, the nominated mandatory and excluded sensor locations, the ranking criteria/linear-based function coefficients, and details of final sensor set selection. A full report is also automatically generated that includes all of the above for every design iteration conducted plus a log of every potential sensor set generated during the design process.

IV. Conclusion

This paper has outlined the basis for development of a suite of software tools to support PHM design and optimisation. The software uses both dynamic simulation and conceptual mapping techniques to facilitate rapid generation of failure knowledge-bases which can be used to predict failure modes and effects, optimise system monitoring designs and develop model-based diagnostics.

References

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

[2] Stone, R. B. and K. L. Wood (2000). "Development of a Functional Basis for Design." Journal of Mechanical Design 122(4): 359-370.

[3] MIL-STD-1629A, “Procedure for performing a failure mode, effects and criticality analysis”, 1980, Department of Defense, Washington DC

[4] Borgovini, R., Pemberton S, and Rossi, M. (1993) Failure Mode, Effects and Criticality Analysis (FMECA), Report CRTA-FMECA, Reliability Analysis Center, Rome USA