The language of FMEA: on the effective use and reuse of FMEA data

S. D. Rudov-Clark and J. Stecki

PHM Technology Pty Ltd
1/15 Pickering Rd Mulgrave 3170 VIC

Abstract

Practical uses of Failure Modes and Effects Analysis (FMEA) range from the identification of potential design defects and safety hazards, to maintenance planning, diagnostics and Prognostics and Health Management (PHM). According to the broadly accepted standard for FMEA, MIL-STD-1629A, a successful FMEA is one that conducted in a timely manner, so that the results can be used to identify potential design flaws and thus mitigate the risk or criticality of a system in its early stages. This paper reviews and compares the current usage of terminology to define functions and failures in accordance with those defined in MIL-STD-1629A. It is argued that inconsistencies in the interpretation and application of such terms has reduced the effectiveness of FMEA, and that a standardised functional and failure taxonomy can alleviate this problem. The functional and failure taxonomies proposed in this paper were created for use in an automated FMEA generation software product. The taxonomy is used to build a functional system model upon which Fuzzy Cognitive Maps are based to predict the system-wide effects of failure. The challenge was to develop a taxonomy that would achieve consistent, repeatable results in automated functional FMEA and enable a wider application and sharing of the FMEA results. The intention was to create a software package that could be used in a concurrent engineering environment, and provide reusable, shareable models for use across Design, Reliability and Maintenance (R&M) and Prognostics and Health Management (PHM) departments within an organisation.

Keywords: FMEA, PHM, reliability, failure, taxonomy, functional definition

Introduction

Failure Modes and Effects Analysis (FMEA) is a powerful tool for evaluating and enhancing system reliability that is used in a wide variety of industries including aerospace, automotive, medical, mining, offshore and power generation. Practical uses of FMEA include identification of potential design defects and safety hazards, maintenance planning and trouble-shooting. A FMEA report is often required to demonstrate compliance with safety and quality requirements such as ISO 9001, QS 9000 and ISO/TS 16949 [1]. When properly maintained and updated, the FMEA can be used as a Knowledge Base for Fault Detection and Isolation (FDI) and Condition Based Monitoring (CBM) applications.

With such a broad range of potential applications, different approaches to FMEA have developed. As a result, both the focus of a FMEA and its terms of reference vary widely depending on the application, as outlined in Table 1. For example, in Design FMEA, the focus of the FMEA is the product itself, therefore the failure terminology relates to the performance, upkeep and safety of the product. By contrast, Process FMECA focuses on the raw material transformation, part/component production and system assembly. Failures and their effects relate specifically to the production process and the quality control systems in place [2].

Table 1: FMEA approaches

FMEA type	**Modelling approach**	**Applications/benefits**
Conceptual design	Functional	Focus on product Risk management through design
Detail design, system upgrades	Hardware	Focus on product Risk management through design identification of system monitoring requirements
Process	Functional/hardware	Focus on process Risk management through design Improve reliability via process controls
Service/operational	Hardware	Focus on product Improved Fault Detection and Isolation (FDI) Risk management through planned risk mitigation strategies (developed in design phase)

This research examines the language-related problems encountered by FMEA practitioners and presents a proposed uniform framework for defining functions and failure concepts. The purpose of the research was to support the development of an automated FMEA software tool, Maintenance Aware Design Environment (MADe), that aims to integrate FMEA with system health management.

Functional taxonomy for FMEA

The key stages of the FMEA process are identified in MIL-STD-1629A [3]; at each stage the issue of language as a hindrance to its effective application has been cited by a variety of researchers and practitioners. In this section the results of a literature review on this issue is presented. While the challenges of effective FMEA are not restricted to taxonomy alone, the following review focuses on problems which can be ameliorated, if not eliminated, by the adoption of a uniform language structure.

(i) System definition: Functions of elements and interface elements may be described by a functional narrative which is a summary of the operations, expected performances and interactions between system elements. This step is essential to FMEA because the MIL-1629A definition of failure mode emphasises that it represents the deviation of functional output of a failed component/subsystem from its acceptable range. A key problem here is the inherent variation in the natural language used to define functions, which is prone to ambiguity [4]. This means that when FMEAs are reviewed by outside parties or after time has passed, the information is unintelligible [5]. This hampers the use of FMEA results throughout the product life-cycle or for reuse on other system designs.

Identified requirements: a restricted number of different terms used to define functions; clear, unambiguous definitions for terms used in functional definitions.

(ii) Identification of Failure Modes and Effects: FMEA requires the identification of all potential item and interface failure modes and their effect/s on the item, and ultimately the system, mission, operation or process. A major shortcoming of the way FMEA is currently done relates to imprecise or inconsistent definition of failure terms and the coverage of failure modes. If failure modes are not clearly defined, engineers and designers will use their own judgement as to what is considered a failure, which can lead to inconsistencies in the analysis [6]. Further, imprecise distinctions are often made between failure modes, causes and effects. This confusion can lead to problems when assigning criticality rankings for FMECA [7]. Kmenta, Cheldelin and Ishii point to the lack of clear guidelines for identifying failures as the reason behind the problem [8].

Identified requirements: a consistent, uniform method for defining failure modes, definitions that provide clear distinction between failure concepts such as failure mode, mechanism, fault, cause and effect.

(iii) Reporting of FMEA results: Vast amounts of data can be generated by a FMECA, the challenge being to report such data in a way that helps to improve system/operation safety and reliability [3]. The MIL-STD provides a template for presenting a concise summary of the FMECA results in tabular form however the report is incomplete if it doesn't include details of the failure mechanism [9].

Identified requirements: inclusion of a brief statement of cause/s, mechanism and local effect/s for each failure mode reported in a FMEA, access to more detailed information on the exact nature of the failure mechanism via a system failure database.

(iv) Implementation of FMEA recommendations: Effective implementation of FMEA requires efficient management of the data generated. At all stages of the product life-cycle, a FMEA report should be amenable to interpretation and adaptation. For instance, for rapid updates during design iterations and system upgrades, or for transitioning the analysis from a conceptual, functional level to a detailed hardware level. Further, the FMEAs supplied by OEMs must be presented in a way that can be understood by the integrator and collated for use in system-level FMEA. The FMEA should allow field reports (example, FRACAS) to be readily incorporated into the FMECA for real-time updates during system operation.

Identified requirements: A system failure Knowledge-Base that is easily updatable – requires some level of automated data-management, uniform taxonomy so that all users are speaking 'the same language; a taxonomy that can be used for both functional and hardware FMEAs.

Development of the made functional taxonomy

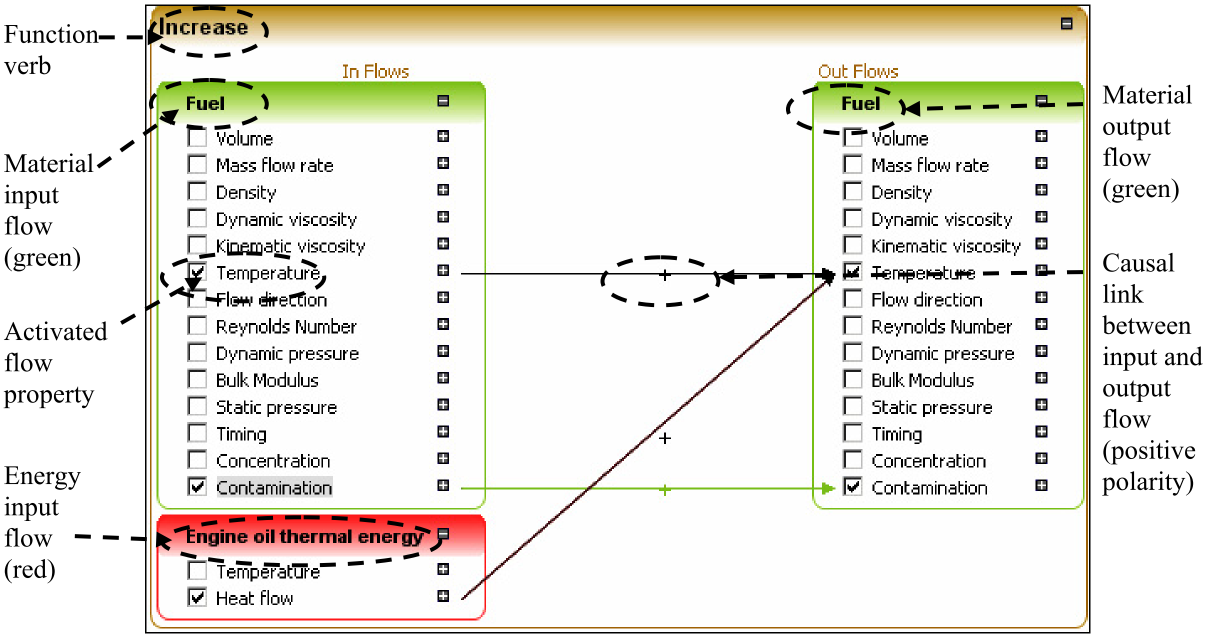
A variety of functional ontologies exist for the purposes of design and problem solving. Chittaro, Guida, Tasso and Toppano's extensive review of functional ontologies classifies them according to two major schools of thought: state-based and flow-based ontology [10]. The state based method represents a function in terms of behaviour states, for example, the Function-Behaviour-State modelling developed by Umeda, Kondoh, Shimomura and Tomiyama [11]. Using the flow-based method, a function is separated from its purpose and treated as a relation between input and output of energy. This method is more widely adopted due to its compatibility with the bond graph method of dynamic analysis, and it lends itself to the development of a basic, generic terminology of 'primitives' for defining the function. Chittaro points out that a research into flow-based functional taxonomies have independently produced similar sets of primitive libraries, which suggests that the flow-based approach is more universal, and likely to promote consistent interpretation.

In this research, a flow-based functional language developed by Stone and Wood [12] was selected as the basis for defining system functions and functional failure modes. This classification provided a high degree of resolution for both function verbs and flow nouns, and was suited to a wide variety of applications. Stone et al compared and reconciled their taxonomy with the NIST taxonomy developed by Pahl and Beitz [13], which further recommended it as suitable for application to automated, functional FMEA. Using the Functional Basis developed by Stone et al, function verbs are classified in a tree-structured list from the most general classifications (there are eight of these) down to specific terms and their synonyms. Similarly flow nouns are classified under the basic classes of material, energy and signal.

A comparison between the functional ontology developed by Stone and Wood [12] and the Function/Action Hierarchy developed Malin and Throop [14] revealed that Malin's ontology provided a high degree of resolution but a narrower range of concepts as it was limited to the aviation industry. It was found that the structure of the Stone classification system was more suited to application to a wide variety of engineering applications, including aerospace, mining and offshore. The Malin list was therefore incorporated into the Stone and Wood taxonomy to complement the list of synonyms for function verbs. The results of this comparison are shown in Appendix A.

The functional representation system developed by Kirschman, Fadel and Jara-Almonte et al extend the concept of flow-based functional representation to include adjectives which represent a measurable characteristic of the input and output flows [15]. When stated in the form [(input) to (output)], Kirschman et al's functional description indicates the direction of the function, plus the adjective goes one step towards specifying the performance parameters, i.e.: the performance limits of the function. This is particularly useful for application to FMEA where a failure often relates to the deviation of a flow's properties from outside its nominal limits. The advantages of this method were adopted for use in MADe by developing a list of 'flow properties' – these are qualities of the material, energy and signal flows that may be used to define performance, or failure, of a function. Further, in MADe a function is defined by creating a causal link between the input and output flow which can define the direction of the function (from input to output) and the polarity of the causal relationship (i.e.: increasing input will increase or decrease the output). The strength of the causal relationship is assigned a value from +1 to -1, with null value indicating no change to the output flow. Fig. 1 shows the functional definition of a heat exchanger which takes as input fuel and thermal energy to output fuel at an elevated temperature.

The primary function of the heat exchanger, which is to increase the fuel temperature, is displayed and reported in MADe as a text string as follows: 'Increase fuel temperature'.
Due to the structure of a MADe functional model, it is necessary to provide a continuous causal path for all flow properties that propagated through a system. As a result other flow parameters such as contamination are present in the functional description, although these do not relate to the primary function of the heat exchanger. In this example, the heat exchanger, when functioning normally, does not increase or decrease the level of contamination in the fuel flow, hence the strength of the causal relationship between the input fuel contamination and the output fuel contamination would be zero. Similar sub-functions can be constructed for the pressure or flow rate of the fuel (as an energy flow).



Fuel (In Flows)
☐ Volume
☐ Mass flow rate
☐ Density
☐ Dynamic viscosity
☐ Kinematic viscosity
☑ Temperature
☐ Flow direction
☐ Reynolds Number
☐ Dynamic pressure
☐ Bulk Modulus
☐ Static pressure
☐ Timing
☐ Concentration
☑ Contamination

Engine oil thermal energy
☐ Temperature
☑ Heat flow

Fuel (Out Flows)
☐ Volume
☐ Mass flow rate
☐ Density
☐ Dynamic viscosity
☐ Kinematic viscosity
☑ Temperature
☐ Flow direction
☐ Reynolds Number
☐ Dynamic pressure
☐ Bulk Modulus
☐ Static pressure
☐ Timing
☐ Concentration
☑ Contamination

Fig. 1. Functional definition for a heat exchanger using the MADe method

Definition of functional failure modes

According to MIL-STD-1629A, a failure mode should describe the way an item fails to fulfil its function:

"3.1.14 Failure mode. The manner by which a failure is observed. Generally describes the way the failure occurs and its impact on equipment operation."

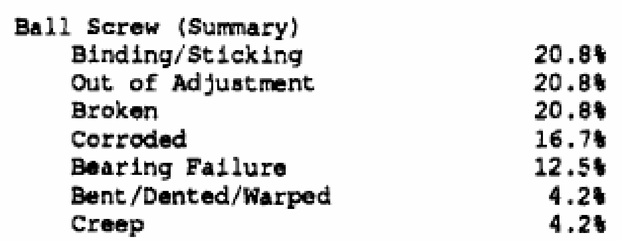
"4.3.5 Failure definition. The contractor shall develop general statements of what constitutes a failure of the item in terms of performance parameters and allowable limits for each specified output." [3]

The key requirements for defining a failure mode are summarised below:

- relates to how failure is observed
- describes the way failure occurs
- describes the impact of failure on equipment
- relates to performance parameters of the item

From the above list it is apparent that the definition of failure mode demands the inclusion of a lot of information. Clearly, terms such as 'broken' or 'failed' do not fulfil all the requirements stated above, and yet such terms are often used to describe failures - an example is shown in Fig. 2. On the other hand, it is unrealistic to expect that a single term for 'failure mode' could encompass information relating to both functional and physical aspects of failure.

The data in Fig. 2 is an extract from the Reliability Analysis Center, now Alion System Reliability Center, which provides current reliability data for a wide range of components and parts. The RAC point out that a failure should refer to external signs of failure rather than the internal mechanics of failure: yet we see that the list of failure modes for a ball screw consists mainly of descriptions of physical faults, such as corroded, broken and bent.



Ball Screw (Summary)	
Binding/Sticking	20.8%
Out of Adjustment	20.8%
Broken	20.8%
Corroded	16.7%
Bearing Failure	12.5%
Bent/Dented/Warped	4.2%
Creep	4.2%

Fig. 2: Summary of failure modes for a ball screw ([16]

To overcome this problem the RAC advises:

"It is important to make the distinction that a failure mode is an 'observed' or 'external' effect so as not to confuse failure mode with failure mechanism". [16]

Later versions of the reliability data published by Alions has sought to rectify this problem by separating 'mechanisms' and 'failure modes' and adjusting the failure mode ratios accordingly, however detailed instructions on how to discriminate between them are lacking, and the MIL-STD-1629A does not provide any useful instructions on how to achieve this.

The MIL-STD requirements for describing failure modes demand definition at a number of different levels of abstraction. Fig. 3 illustrates this point by comparing the requirements of failure definition against the Abstraction Hierarchy defined by Rassmussen [17], and further developed by Lind [18].

In this paper (section 2(ii)), it has been argued that another common problem with the practice of FMEA is the treatment of the terms 'failure mode' and 'failure cause' as synonyms. The MIL standard defines failure cause as follows:

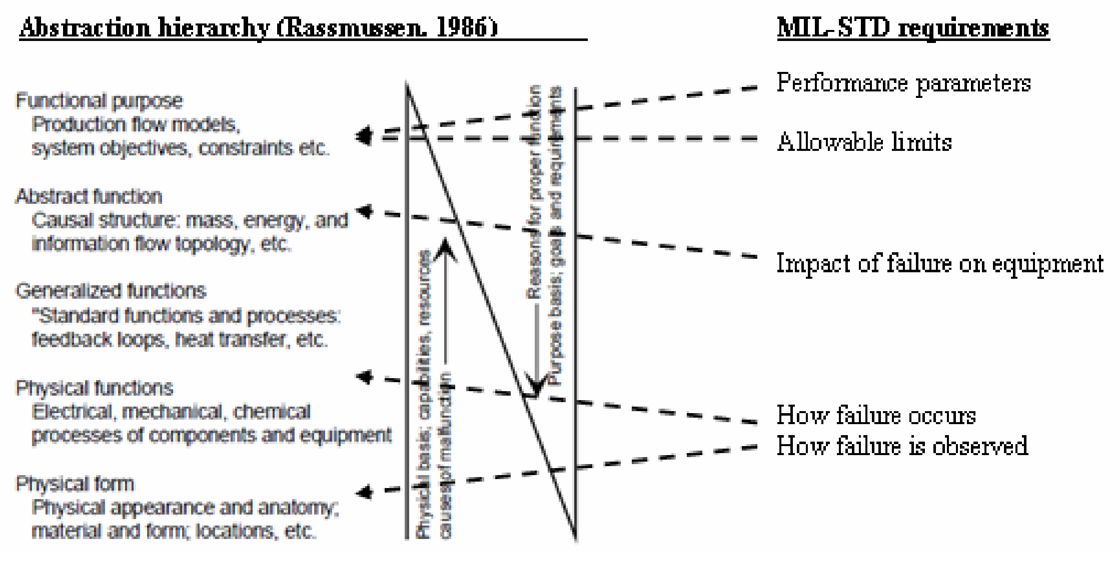


Fig. 3.: MIL-STD-1629A requirements for failure definition and Rassmussen's abstraction hierarchy

Table 2: Definitions of failure concepts

Concept	Definition	MADe modelling approach	Level of abstraction
Failure mode	the way in which an item fails to fulfil its function.	stated as a combination of function verbs, with flow nouns to define inputs and outputs causal links between input and output flows define the causality of the function	highest – it is the state of a functional flow, not the physical state of the item.
Fault	the damaged state of a system element that renders it unfit to fulfil its function.	modelled in a 'failure diagram' using a standardised list of descriptors for physical damage	lowest – it defines the physical state of the item.
Mechanism	the physical process of degradation.	modelled in a 'failure diagram' using a standardised list of failure mechanism terms	lowest
Cause	the abnormal state of input, loading or environment that leads to the degradation of an item.	modelled in a 'failure diagram' using a standardised list of causes	lowest

"3.1.12 Failure cause. The physical or chemical processes, design defects, quality defects, part misapplication, or other processes which are the basic reason for failure or which initiate the physical process by which deterioration proceeds to failure." [3]

The challenge is therefore to provide a definition for 'failure mode' which relates to the non-fulfillment of function (highest level of abstraction), can be expressed in terms of performance parameters (second highest level of abstraction), can be physically observed (lowest level of abstraction), and used to deduce its effect on other equipment (second highest level of abstraction). This must be achieved while preserving the distinction between internal mechanics of failure and externally observed functional failure mode. It is proposed that such a set of objectives is best met using multiple, distinct failure concepts. In MADe we have separated 'failure mode' from lower level of abstraction concepts as shown in Table 2.

Fig. 4 provides an example of a failure diagram generated in MADe, in which the physical failure concepts associated with two parts are linked via causal connections to the higher level functional failure concepts for the two parts working in unison within a component. A detailed description of the physical failure concept taxonomy and how they are mapped in the failure diagram is beyond the scope of this report. Here we focus on the definition of 'failure mode' and its application to FMEA. The salient feature of this terminology is the clear distinction between lower levels of abstraction – which relate to the physical state of a system element, and its functional failure mode, which relates exclusively to its performance parameters, via changes to its output flows.

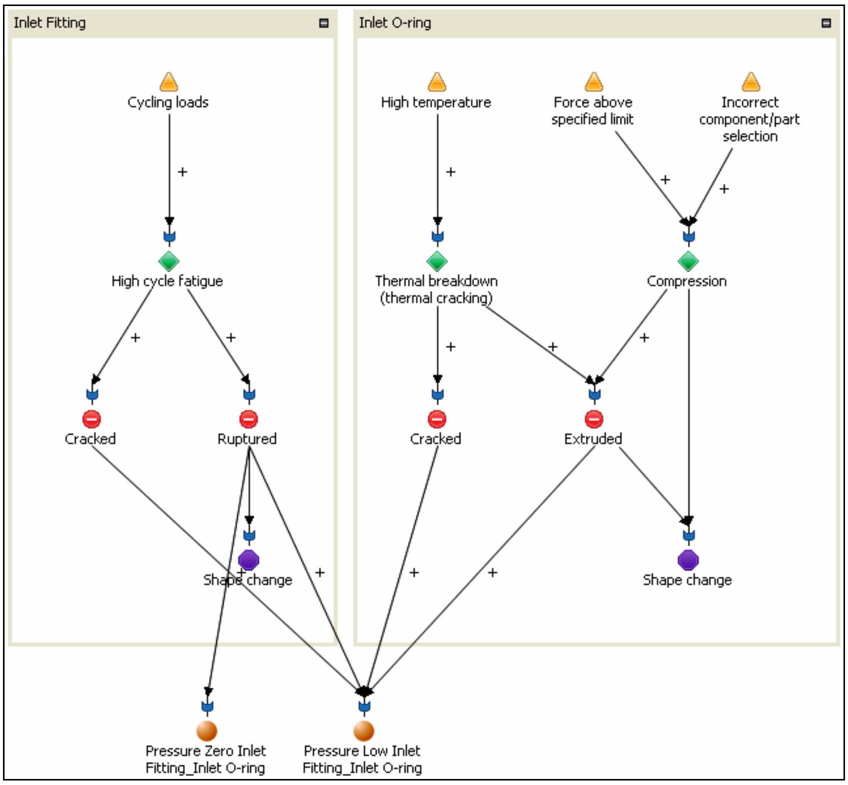


Fig. 4: Causal links between physical and higher level functional failures (bottom of image)

Table 3: Summary of input flows for FDS system model

Input flow			**Source (external)**	**Target component**
Class I	**Sub-class**	**Type**		
Energy	Thermal	Heat flow	Engine oil	Heat exchanger
Material	Liquid	n/a	Fuel tank	Heat exchanger
Energy	Mechanical	Rotational	Engine gearbox	Fuel pump assembly
Energy	Mechanical	Rotational	Engine gearbox	Fuel pump assembly
Energy	Hydraulic	Pressure	Fuel	Heat exchanger
Signal	Control	Discrete	FADEC	Fuel metering unit
Signal	Control	Discrete	Shutoff valve	Fuel metering unit
Signal	Control	Discrete	Fire switch	Fuel metering unit
Signal	Control	Discrete	Start level	Fuel metering unit

Automated FMEA analysis using a standardised functional and failure language

A functional model is used to represent a system and its elements in terms of their functional interactions. This is useful for complex, nonlinear systems or systems at the conceptual design phase. In a MADe functional model, component interactions are represented by the exchange of functional flows of material, energy and signals, as shown in Fig. 5.

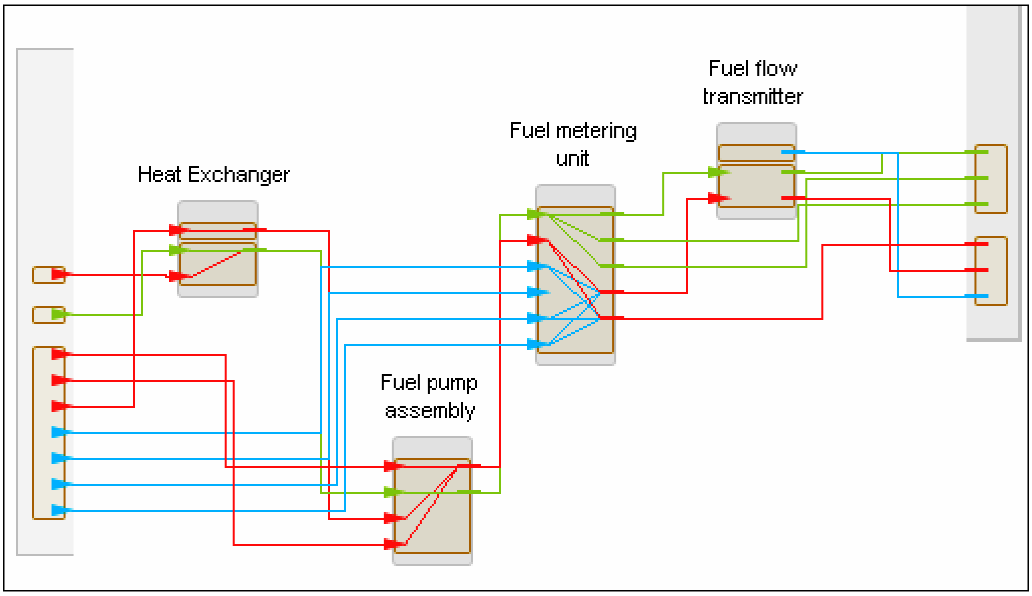
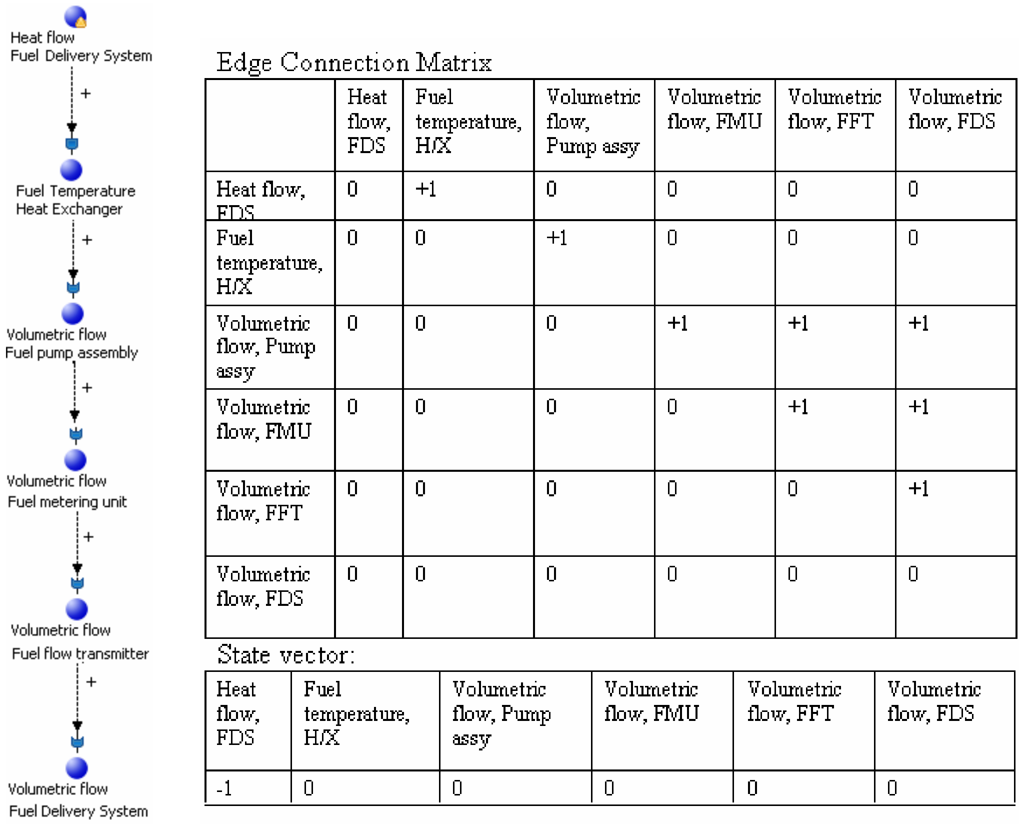


Fig. 5: Functional model of a generic Fuel Delivery System (FDS)

Fig. 5 presents the system structure at highest level of indenture in a generic Fuel Delivery System (FDS). System indenture levels are connected by linking the input flow and output flows between levels and this allows failure effects to be propagated from part level through to system level to determine 'next' and 'end' effects of failure modes. The material, energy and signal flows between components are colour coded. The MUX-type bars at either side of the model schema represent the interface between the fuel delivery system and other systems within an aircraft. For example, the system takes, as input flows, the materials, energies and signals listed in Table 3. The flows are listed in this table in the order that they appear on the model's input MUX-bar (Fig. 5) from top to bottom.

In MADe, a Fuzzy Cognitive Map (FCM) is automatically constructed based on a MADe functional system model, in which functional concepts are connected by the causal linkages in each component. Functional failure modes are introduced to the system via the lower level of abstraction 'failure diagrams', which model the physical processes that lead to a functional failure mode. The impact of each of a component's Functional Failure Modes (FFMs) is simulated by propagating the erroneous output flow through the causal links in the system FCM.

In Fig. 6, part of the Fuzzy Cognitive Map for the Fuel Delivery System is shown. A failure path has been initiated by the input of low thermal energy (heat flow) to the heat exchanger, from the engine oil. A causal path exists between this initial failure event and the final system-level failure of the FDS – 'volumetric flow low" – i.e.: the volumetric flow of the fuel (modelled as an energy flow – hydraulic energy). All such causal paths are identified via a path finding algorithm which steps through all functional causal connections within the system to create an Edge Connection Matrix.



Edge Connection Matrix

	Heat flow, FDS	Fuel temperature, H/X	Volumetric flow, Pump assy	Volumetric flow, FMU	Volumetric flow, FFT	Volumetric flow, FDS
Heat flow, FDS	0	+1	0	0	0	0
Fuel temperature, H/X	0	0	+1	0	0	0
Volumetric flow, Pump assy	0	0	0	+1	+1	+1
Volumetric flow, FMU	0	0	0	0	+1	+1
Volumetric flow, FFT	0	0	0	0	0	+1
Volumetric flow, FDS	0	0	0	0	0	0

State vector:

Heat flow, FDS	Fuel temperature, H/X	Volumetric flow, Pump assy	Volumetric flow, FMU	Volumetric flow, FFT	Volumetric flow, FDS
-1	0	0	0	0	0

Fig. 6: Partial FCM of the Fuel Delivery System model and Edge Connection Matrix

The Edge Connection Matrix method involves a matrix of n by n concepts, with the cells containing the strength and polarity of each causal connection: i.e.: values ranging from +1 to -1. A state vector of dimension n represents activated concepts. A failure mode is activated by assigning a value from +1 and -1 to the appropriate cell in the state vector. By multiplying the Edge Connection Matrix by the state vector, a new state vector is calculated which will signify any concepts that have been switched on by the activated failure mode by the presence of a non-zero value in the corresponding cell.

The example in Fig. 6 shows a state vector in which the heat flow to the FDS (from the engine oil) has been perturbed down. This means that less than nominal thermal energy is being transferred to the heat exchanger from the engine oil. This state vector, when multiplied by the Edge Connection Matrix until a steady state is reached, provides the steady state response of every component in the system. Steady-state is reached when consecutive matrix multiplications produce no change in the state vector, or when a repeated cyclic pattern occurs.

It should be noted that the causal relationship between fuel temperature and volumetric flow through the fuel pump assembly was automatically generated by lower level of indenture within the fuel pump assembly. Within the assembly, the fuel filter component possesses a failure diagram in which low temperature fuel is a cause for icing, which clogs the filter. This is causally connected to the functional failure mode ‘low volumetric flow’ within the component failure diagram. Lower level failure diagrams, which relate to physical degradation of parts and components, will be discussed in a subsequent paper on this topic. The benefit of propagating failures through the system model at a functional level only is that conceptual designs can be analysed so that FMEA helps inform the design process. This signifies a fundamental shift from FMEA as a ‘check-box’ item at the completion of a design to FMEA as a design optimisation tool, specifically for risk mitigation and maintenance management.

To generate a FMEA report, MADe automatically activates every possible failure mode via

- causes provided in the failure diagrams, and
- causes generated by failed input flows to the system

The functional flows between components, combined with lower level of abstraction failure diagrams, are used to generate an Edge Connection Matrix which represents the causal relationship between concepts at every level of indenture in a functional system model. The Edge Connection Matrix method is used to calculate the final state of the system due to activated failure mode. The results are used to determine the next and end-effects for every failure mode - for FMEA reporting and to serve as a system failure database for other system Health Management activities.

One application of the system failure database in MADe is sensor set design and optimisation. The database provides the system response to a failure mode at every location on the system. This information is used to develop unique sets of ‘symptoms’ for each failure mode. Differentiating between the symptoms of each failure modes, sets of sensor locations can be identified that will enable discrimination between failure modes. Sensor sets can be optimised by the number, weight, cost or reliability of the sensors. The sensor set design and optimisation tool can also be used to provide coverage analysis, which is useful for assessing the coverage afforded by pre-existing sensors on a system.

Conclusion

The development of a functional taxonomy for FMEA applications has led to the segregation of failure concepts according to their position within the abstraction hierarchy. By separating physical failure concepts from functional failure concepts, the following objectives have been achieved:

- a standardised approach to reporting failure modes in FMEAs
- the ability to store non-functional failure concepts within a context-independent failure diagram
- a taxonomy that can be used in FCM analysis for automated failure analysis
- the ability to provide detailed information about failure processes, while maintaining a rigorous level of standardisation and conciseness in the FMEA report

The taxonomy has enabled the development of a computational approach to propagating failure modes through a functional system model, to reduce the cost and time of generating FMEAs while improving repeatability, reliability and the ability to share and reuse FMEA information.

Acknowledgement

The authors wish to thanks Stone and colleagues for permission to use their functional taxonomy and adapt it to the requirements of automated FMEA.

References

1. Bajaria, H. J., ‘Effective execution of FMECA’ in Proc. 6th Annual SAU International RMS Workshop and Exposition, Dallas, 1994

2. Smith, M. ‘Failure Mode and Effects Analysis’ Workshop notes, Rev N 980815, Cayman Business Systems, 2002, http://elsmar.com/FMEA/sld001.htm , last accessed 11/11/2008.

3. Anon., ‘Procedures for performing a failure mode, effects and criticality analysis’, MIL-STD-1629A, US Department of Defense, 1980

4. Chandrasekaran, B., Goel, A. Iwasaki, Y. ‘Functional Representation as Design Rationale’ Computer, 26(1), pp 48-56, 1993

5. Stone, R. B., Tumer, I. Y., Van Wie, M., ‘The function-failure design method’ ASME Journal of Mechanical Design, 127, pp 127-397, 2005

6. Bidokhti, N., ‘How to close the gap between hardware and software using FMEA’, in Reliability and Maintainability Symposium, pp 167-172, 2007

7. Kara-Zaitri, C., Keller, A.Z., Barody, I., Fleming, P.V., ‘An improved FMEA methodology’ in Proc. Reliability and Maintainability Symposium, pp 248-252, 1991

8. Kmenta, S., Cheldelin, B., Ishii, K., ‘Assembly FMEA: a simplified method for identifying assembly errors’ in Proc. IMECE’03, November 16-21, 2003, Washington DC

9. Cassanelli, G., Mura, G., Fantini, F., Vanzi, M., Plano, B., ‘Failure-Analysis assisted FMEA’ Microelectronics Reliability, 46(2006), pp 1795-1799, Elsevier 2006

10. Chittaro, L., Guida, G., Tasso, C., Toppano E., ‘Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: a case study in diagnosis.’ in IEEE Transactions on Systems, Man and Cybernetics, 23(6), pp 1718-1751, 1993

11. Umeda, Y., Kondoh, S., Shimomura, Y., Tomiyama, T., ‘Development of design methodology for upgradable products based on function-behaviour-state modelling’, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, v. 19, pp 161-182, Cambridge University Press, 2005

12. Stone, R. B., Wood, K., L., ‘Development of a functional basis for design’, Journal of Mechanical Design, v. 122, pp. 359, ASME, 2000

13. Pahl, G., Beitz, W., ‘Engineering Design: A Systematic Approach’, Springer, 1996

14. Malin, J. T., Throop, D. R., ‘Basic concepts and distinctions for an Aerospace ontology of functions, entities and problems’ in Proc. IEEEAC, pp 1-18, 2007

15. Kirschman, C. F., Fadel, G. M., Jara-Almonte, C. C., ‘Classifying functions for mechanical design’, in Proc. The 1996 ASME Design Engineering Technical

Conference and Computers in Engineering Conference, Paper ID 96-DETC/DTM-1504, Irvine, August 18-22, 1996

16. Chandler, G., Denson, W. K., Rossi, M. J., Wanner, R. 'Failure Mode/Mechanism Distributions', DTIC Research Report ADA259655, 1991

17. Rasmussen, J., 'Information Processing and Human-Machine Interaction', Elsevier Science Publications, New-York, 1986, 228 pages

18. Lind, M., 'Making sense of the abstraction hierarchy', in Proc. Proceedings of the seventh European Conference on Cognitive Science Approaches to Process Control, pp 195-200, Villeneuve d'Ascq, 21-24 September, 1999

Appendix A Malin and Throop vs. Stone and Wood functional taxonomies

Stone and Wood Functional Taxonomy				Malin and Throop terms	
Primary class	Secondary class	Definition	Synonyms	Class/es	Terms
Branch		To cause a material or energy to no longer be joined or mixed.	divide, diverge, bifurcate, separate,	Change assembly	disassemble, disorder
	Separate	To isolate a material or energy into distinct components. The separated components are distinct from the flow before separation, as well as each other.	Isolate, sever, disjoin	Change mixture, Arrange/put	separate, clean, break down, isolate
	Divide	To split up a flow into parts or to classify distinct parts of a flow.	Detach, isolate, release, sort, split, disconnect, subtract	Change assembly, Arrange/put, Change shape	disassemble, disconnect, split, rupture
	Extract	To draw, or forcibly pull out, a flow.	Refine, filter, purify, percolate, strain, clear		
	Remove	To take away a part of a material from its prefixed place.	Cut, polish, sand, drill, lathe	Use, change shape	forego, waste, eat away
	Refine	To reduce a material or energy such that only the desired elements remain.			
	Distribute	To cause a material or energy to break up. The individual bits are similar to each other and the undistributed flow.	Diverge, scatter, disperse, diffuse, dispel, resist, dissipate		
Channel		To cause a material or energy to move from one location to another location	conduct, convey, transmit, spread, pilot, guide, steer, route, transfer	Hold	channel
	Import	To bring in an energy or material from outside the system boundary.	Input, allow, form entrance, capture	Shift/distribute	insert
	Export	To send an energy or material outside the system boundary.	Eject, dispose, remove, emit, empty, destroy, eliminate	Destroy/injure	obliterate, burn, deprive
	Transfer	To shift, or convey, a flow from one place to another.	Carry, deliver	Shift	transfer, shift, move

	Transport	To move a material from one place to another.	Lift, move	Hold, Shift/distribute	carry, move smoothly
	Transmit	To move an energy from one place to another.	Advance, conduct, convey	Hold	drop
	Guide	To direct the course of an energy or material along a specific path.	Direct, shift, switch, straighten, steer		
	Translate	To fix the movement of a material (by a device) into one linear direction.	Move, relocate		
	Rotate	To fix the movement of a material (by a device) around one axis.	Turn, spin		
	Allow DOF	To control the movement of a material (by a force external to the device) into one or more directions.	Constrain, unlock, unfasten	Hold, change mobility, control	give way, destabilise, free, allow
Connect		To bring two or more energies or materials together.	couple, associate, link, correlate, join, associate, link, combine, bind	Arrange/put	collide
	Couple	To join or bring together energies or materials such that the members are still distinguishable from each other.	Associate, connect		
	Join	To couple flows together in a predetermined manner.	Assemble, fasten	Change assembly, Connect	
	Link	To couple flows together by means of an intermediary flow.	Attach		
	Mix	To combine two materials into a single, uniform homogeneous mass.	Combine, blend, add, pack, coalesce	Change mixture	interchange, combine, contaminate,
Control Magnitude		To alter or govern the size or amplitude of material or energy.			
	Actuate	To commence the flow of energy or material in response to an imported control signal.	Enable, start, initiate, turn on	Control	allow
	Regulate	To adjust the flow of energy or material in response to a control signal, such as a characteristic of a flow.	Control, equalize, limit, maintain	Change energy, Control	change force, maintain, preserve, ensure

	Increase	To enlarge a flow in response to a control signal.	Allow, open	Change energy	increase
	Decrease	To reduce a flow in response to a control signal.	Close, delay, interrupt	Change energy	reduce
	Change	To adjust the flow of energy or material in a predetermined and fixed manner.	Adjust, modulate, clear, demodulate, invert, normalize, rectify, rest, scale, vary, modify	Control	rectify, save, restore, accommodate, rework, reset
	Increment	To enlarge a flow in a predetermined and fixed manner.	Amplify, enhance, magnify, multiply	Change amount, Control	produce, increase, copy, maximise
	Decrement	To reduce a flow in a predetermined and fixed manner.	Attenuate, dampen, reduce	Control	mitigate, minimise
	Shape	To mold or shape a material.	Compact, crush, compress, pierce, deform, form	Change shape	shape, compress, expand, crease, cut, penetrate, smooth, roughen, break open
	Condition	To render a flow appropriate for the desired use.	Prepare, adapt, treat		
Stop		To cease, or prevent, the transfer of a material or energy.	End, halt, pause, restrain, break, cease, block, limit, arrest, reject, barrier, choke, delay, pause, interrupt, suspend, close, finish, close	Change service availability, control	withhold, undo
	Prevent	To keep a flow from happening.	Disable, turn off		
	Inhibit	To significantly restrain a flow, though a portion of the flow continues to be transferred.	Shield, insulate, protect, resist	Control	avoid, withstand, guard
Convert		To change from one form of energy or material to another. For completeness, any type of flow conversion is valid. In practice, conversions such as convert electricity to torque will be more common than convert solid to optical energy.	Transform, liquefy, solidify, evaporate, condense, integrate, differentiate, process, create, decode, encode, generate, digitize, shift, swing, varying, adjust, switch, convert, translate, variate, adapt	Convert	change phase, change hardness/strength, cook, digest
Provide		To accumulate or provide material or energy.			

	Store	To accumulate material or energy.	Accumulate, stock, keep, group, pack, bunch,cache, bundle, concentration	Hold	Store
	Contain	To keep a flow within limits.	Capture, enclose		
	Collect	To bring a flow together into one place.	Absorb, consume, fill, reserve	Arrange/put	close, cover
	Supply	To provide material or energy from storage.	Provide, replenish, retrieve	Change service availability	provide
	Extract	To draw, or forcibly pull out, a material or energy.		Arrange/put	expose, uncover
Signal		To provide information.			
	Sense	To perceive, or become aware, of a signal.	Feel, determine		
	Detect	To discover information about a flow.	Discern, perceive, recognize		
	Measure	To determine the magnitude of a material or energy flow.	Identify, locate	Inform	Classify, measure
	Indicate	To make something known to the user.	Announce, show, denote, record, register	Inform	indicate, record
	Display	To reveal something about a flow to the mind or eye.	Emit, expose, select		
	Track	To observe and record data from a flow.	Mark, time	Control, inform	monitor
	Process	To submit information to a particular treatment or method having a set number of operations or steps.	Compare, calculate, check	Inform	convert, evaluate, check, test, validate
Support		To firmly fix a material into a defined location, or secure an energy into a specific course.			
	Stabilize	To prevent a material or energy from changing course or location.	Steady	Hold	support, stabilise, secure
	Secure	To firmly fix a material or energy path.	Attach, mount, lock, fasten, hold, place, constrain, fix	Arrange/put	immobilise
	Position	To place a material or energy into a specific location or orientation.	Orient, align, locate	Change position, arrange/put	position