Proceedings of the International Manufacturing Conference, IMC19 (pp. 263-272).
Queen's University, Belfast, UK, 28-30 August 2002.

FAILURE MODE AND EFFECT ANALYSIS GENERATION FOR CONCEPTUAL DESIGN

Ping Chow Teoh and Keith Case

Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK.

ABSTRACT

Failure Mode and Effect Analysis (FMEA) is a widely used concurrent engineering tool for quality improvement and risk assessment. However, many shortcomings have hindered its effectiveness. The research described here aims to contribute to the implementation of FMEA in conceptual design by eliminating some of these shortcomings. The focus of the work is on the information modelling of FMEA knowledge, and the emphasis is on the avoidance of additional workload for the designer and the encouragement of knowledge reuse. A relational data model has been created to support the automatic generation of the FMEA. This automatic generation replaces the traditional brainstorming process for FMEA report creation. Inputs of failure reports from the factory floor are used for FMEA generation. As an alternative approach, designers can provide the characteristics of the components of their design to generate the FMEA. The user has the final decision on whether the FMEA generated are to be recorded as the final FMEA report. Prototype software has been created to demonstrate the above capabilities. The data model is also intended to support the viewpoints of multiple users, namely, the product designer, the field engineer, the process engineer and the maintenance engineer. Further research is in progress.

KEYWORDS: FMEA, knowledge reused, Concurrent Engineering

1. INTRODUCTION

The rapid changes in technology and market requirements present manufacturing organizations with tremendous challenges. Many companies are forced to change their normal way of running the business in order to survive the fierce competition. The common strategies employed to cope with the evolving business environment include adapting new technology, performing organizational restructuring and core processes redesign. Indices such as quality, cycle time and cost are commonly used as measurements against the common objectives for new product introduction, "faster, better and cheaper". One of the major drivers to achieve the above objectives is concurrent engineering [1]. As a result, many tools have been introduced. FMEA is an engineering tools that is adapted to achieve the objectives of concurrent engineering [2].

1.1 Failure Modes and Effects Analysis

BS 5760 Part 5 [3] states that "FMEA is a method of reliability analysis intended to identify failures, which have consequences affecting the functioning of a system within the limits of a given application, thus enabling priorities for action to be set." Basically, FMEA can be classified into two main types: Design FMEA and Process FMEA. Design FMEA deals with design activities, such as the design of products, machines and tooling, whereas process FMEA is concerned with manufacturing processes.

1.2.1 Shortcomings of the Current FMEA

Despite its potential impact and wide usage, there are not many good examples to show the benefit of FMEA to manufacturing organizations. This is due to several weaknesses in the method itself that need to be improved.

Traditionally, FMEA is used in hard copy or spreadsheet format to capture the potential problems of a design or process. The implementation of a highly manual FMEA is a difficult task. FMEA is found to be not user friendly, hard to understand and insufficiently flexible. There is also duplication of information in other documents in the factory. As a result, many companies use FMEA merely to satisfy the contractual requirements of their customers [4]. In other words, it has become extra and non value-added work to the engineers. Users may find FMEA a "tedious and time-consuming activity" [5]. It is especially true when FMEA is used in complex systems with multiple functions [3]. FMEA is often carried late in the design cycle after the design prototype has been built [5]. The changes made at later stages will be very costly, and a poor design concept can never be compensated for at later design stages [6].

According to Wirth et al [7], the method has two fundamental weaknesses. Firstly, there are no guidelines on how to conduct an FMEA. Hence, the methods employed vary from one application to another. Secondly, all FMEA-related information is recorded in natural language, with team members using their own vocabulary. Lee [8], believed the limitation of the FMEA method is due to the unstructured knowledge representation and limited reasoning support. In short, the main problem with FMEA lies in the fact that it is too human dependent and the human is not able to provide a consistent way of creating and maintaining FMEA's.

Hence, it is clear that one of the effective ways to overcome the current shortcomings is to automate the FMEA generation. Many manufacturing companies have their own method of recording previous product or process failures, and this information may be sufficient to generate important elements within a comprehensive FMEA. However, in addition to this knowledge, relevant modelling and reasoning techniques are required to support the automatic FMEA generation method.

2. FMEA MODELLING AND REASONING

2.1 FMEA Modelling

The knowledge in the FMEA needs to be modelled and codified to support the automation process. Knowledge of both structure and the function of the system under consideration are needed for FMEA automation [9, 10]. Hence, functional modelling and structural modelling are often associated with FMEA research.

A functional model describes the intended function or the purpose of a system, and consists of two main components: function and behaviour. The function of a system provides the design intent, whereas the behavior describes how the structure of an artifact achieves its function [11, 12].

A structural model is defined as "the components that make up an artifact and their relationships" [11]. It refers to the configuration of the product or system. It contains information on all the components, entities, sub-processes or sub-systems, together with the interactions among them. A structural model may typically refer to the physical assembly of a mechanical or electrical product (e.g. a car, an engine, or an electrical circuit), or to a software configuration.

2.2 Theory of Technical Systems

Technical Systems Theory is concerned with achieving a transformation with a well-defined intention. According to Hubka [13], a transformation system is defined as "a sum of all elements and influences (and the relationships among them and their environment) that participate in a transformation”. Briefly, the elements of a transformation system consist of an operand, a technical process, a technical system, a human system and an active environment [13]. Conveniently, the transformation system provides a full picture of the relationships among functional and structural models in a design. The function model is represented in the technical process of the transformation system model, while the structural model is directly linked to the technical system of the model. Figure 1 shows the model of the transformation system.

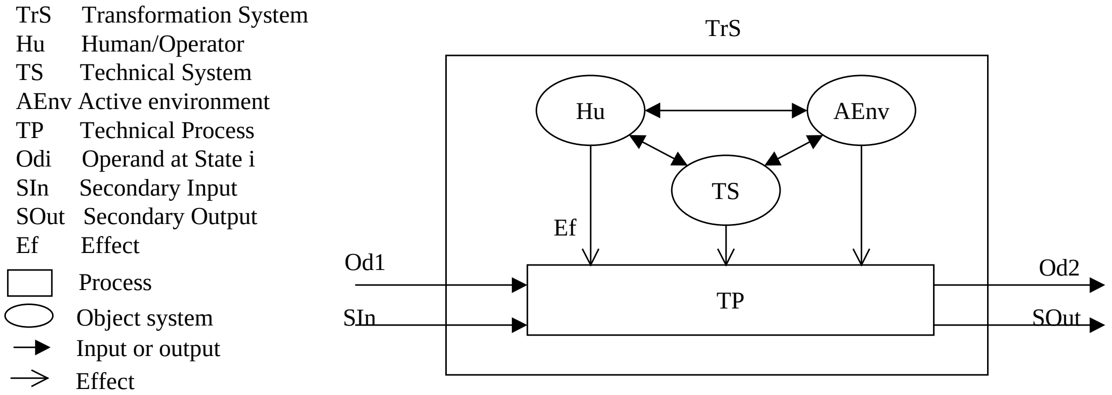


Figure 1 Model of the Transformation System [13]

2.3 FMEA Reasoning

In FMEA automation, brainstorming is replaced by computer reasoning. Hence, the failure cause to effect generation can be carried out by a computer. There are two types of reasoning, i.e. quantitative reasoning and qualitative reasoning.

Quantitative reasoning requires more numerical data. Usually the variables involved are measurable. It is operated based on first principles. For example, a mechanical problem will be reasoned based on the physical laws and the mechanical properties. This type of reasoning can be more refined, but it demands considerable computer memory and computational processing. Hence, it can be very time consuming.

In contrast, qualitative reasoning requires less detail data. The variables used are more descriptive then measurable. For example a force may be described as low, medium, high, etc. instead of given an exact value. This type of reasoning is less precise as compared to quantitative approach, but the process is far simpler.

As the FMEA in this work is intended for early stages of design, numerical detail is usually not available. Furthermore, quantitative reasoning is not the way human analysts tend to tackle problems [14]. FMEA is more the product of human thought processes than of computer reasoning. Hence, qualitative reasoning is more widely used in most FMEA methods.

2.4 Shallow Knowledge vs. Deep Knowledge

A knowledge-based system can be built based on shallow knowledge or deep knowledge. The notion of shallow knowledge and deep knowledge is more of a relative comparison between different sets of knowledge, and there is no clear distinction between the two.

Deep knowledge is used to simulate the actual behaviour of the system based on scientific principles, provided that the sufficient data is available. Numerical simulation is considered as an application of deep knowledge in the reasoning process. If this is not possible, then shallow knowledge will be used. One form of shallow knowledge representation is by heuristic rules in the form of *IF condition*, *THEN conclusion* constructions. The relationship between the condition and the conclusion is totally empirical [14].

The advantage of deep knowledge is that it is in a form of generic principles, thus it can be applied to many cases. However, it can only work on detailed models. For example, to model a car, parts that make up the car assembly need to be considered in detail. The problems faced by this approach include higher development effort, higher computing loads and the consistency of the results. This makes it unrealistic for application in FMEA generation, especially for complex design and manufacturing process.

The advantage of having shallow heuristics is that they are easy to create and fast to conclude if the case matches with the condition. They can be used in a model with high levels of abstraction. For example, a car can be considered as a whole entity for shallow knowledge reasoning. The drawback is that the rules are very narrow, and may not be practical to apply to a complex part, as it may end up having too many rules. The maintainability of such a system can be an issue as the software is directly linked to the engineering system [14].

For the purpose of FMEA automation, despite the drawback mentioned above, shallow knowledge approach is chosen. Success is more likely when using an approach that is less detailed in terms of data, as it can cater for a wider range of product and process designs.

2.5 Current FMEA Research

Considerable research is being carried out for many different FMEA. FLAME is a successful example of FMEA automation for electrical design of automobile systems [5]. GENMech [15] is an attempt to automate FMEA for Mechanical design. Atkinson et al [14] proposed an object-oriented approach for hydraulic systems design. The models suggested by Eubanks et al [10] and Russomanno et al [12] are for a more generic FMEA application. Except for by Eubanks et al [10] and Russomanno et al [12], most of the proposed models are specific in terms of their application. This paper is intended to elaborate the research being carried out along the line of generic FMEA application for both design and process FMEA.

3. PROPOSED FMEA MODEL

3.1 Terminology

The following is a list of terms and notations used in the explanation of the FMEA method:

Part	An object that forms the primary element of a design artifact or process.
Operator	A Part that initiated an interaction with other Parts in a design artifact or process
Operand	A Part that is at the receiving end of an interaction between Parts
Property	A measurement/attribute that represents a characteristic of a Part
State	A condition/value of a Property
Function	The purpose/intent of a design
Generic Function	The purpose/intent of a design that has been categorized into generic grouping
Behaviour	The characteristic or state of a Generic Function
Function Unit	The smallest unit that represents an interaction between an Operator and an Operand with a Function
Model	An assembly of Operators that serve a design purpose/function
Precondition	The relationship between the state of an operator with the behaviour that it has generated

Postcondition	The relationship between the behaviour and the state an operand which is the result of the behaviour
Failure Mode	An undesired Behaviour of a Generic Function
Cause	The State of a Part that causes a State change on other Parts
Effect	The State change that is resulted from a Cause from another Part
[ ]	Indicates that the name in the bracket is an object or a class

3.2 Model Layout

An object-oriented approach is used for FMEA modelling. The model consists of basic classes, such as [Part], [Generic Function], [Function Unit], [Model] and others. A [Part] is characterized by its [Property]s and [State]s. A [Generic Function] can have many [Behaviour] objects associated with it. A [Function Unit] is created based on a [Part] which forms the operator to a [Function], another [Part] which is the operand of the [Function] and a [Generic Function] which describes the [Function] itself.

A series of [Function Unit]s make up a higher level [Function Unit], which is known as the assembly of those units. The operator of the higher level [Function Unit] is the [Model] that go to make up its assembled components, i.e. the operators from the lower level [Function Unit]s. The relationships among the objects can be represented by Figure 2.

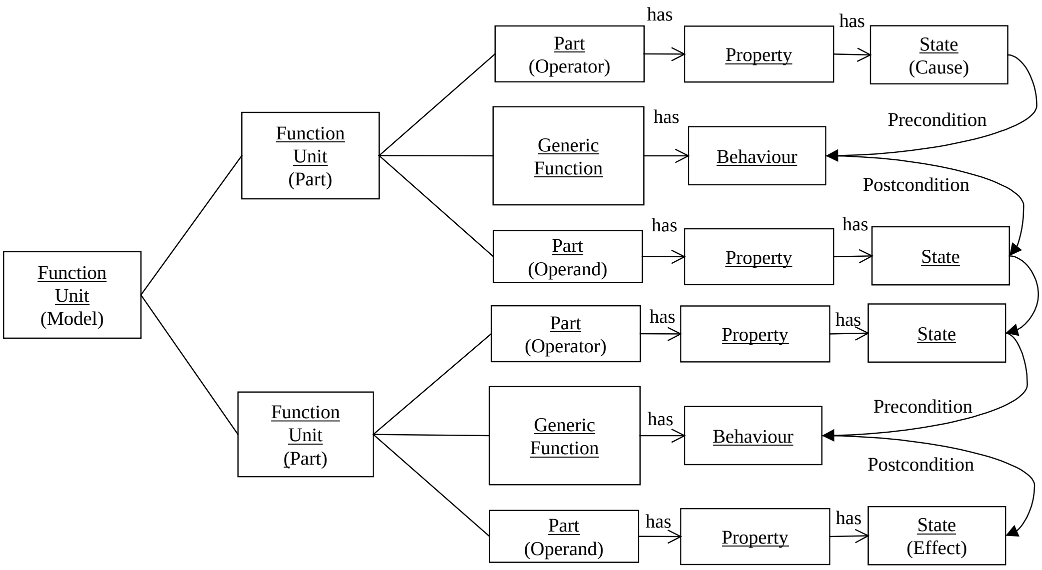


Figure 2 An Example of an Object Layout and their Relationships

3.3 Effect Propagation

The state of the operator will determine the [Behaviour] of the [Generic Function] within a [Function Unit]. This is the precondition relationship. The [Behaviour] will in turn decide the state of the operand within the [Function Unit]. This is termed the post-condition relationship.

The interaction between [Function Unit]s within an assembly is carried out through the [Part] itself. This is because in most cases, an operand of a [Function Unit] is an operator of the next [Function Unit]. Hence, if a state change occurs, the changes will be propagated until the last operand in the system, i.e. the operand that is not used as an operator for the next [Function Unit]. Figure 2 (above) shows as example of effect propagation.

3.4 State Response

The information about various states of a part, and the behaviours corresponding to specific states are stored in the database. If a [Function Unit] is created, the operator, operand and the [Generic Function] involved will be used as the keys to search for the matching [State]s and [Behaviour]s. The selected [State] from the operator is the cause of this [Function Unit], and the effect is defined by the [State] of the operand. Hence, a [Part] is able to act or respond to the system through its distinctive "memory".

3.5 Database

A relational database is used to store the information of the objects as well as the heuristic rules governing the object. This overcomes the difficulty suggested by Atkinson et al [14] that maintainability is a problem if the software is directly related to the engineering system. With rules residing in the database, independent of the software, an engineering change will only affect the information in the database, not the software itself. Furthermore, a database is capable of storing as many rules as needed, resolving the concern on the quantity of rules.

4. APPLICATION IN AUTOMATIC FMEA GENERATION

4.1 FMEA Generation

FMEA generation is accomplished by combining the characteristics of the parts in the part library with the behaviour information. Typical steps that may be involved in an FMEA generation process are as follows:

a) Establish design concept
   During the conceptual design stage, the designers may have a general idea of what technical systems and technical process that they would like to use in order to create a design that will achieve a given intent. An example of a conveyor design concept using the transformation system model [13] is as shown in Figure 3.

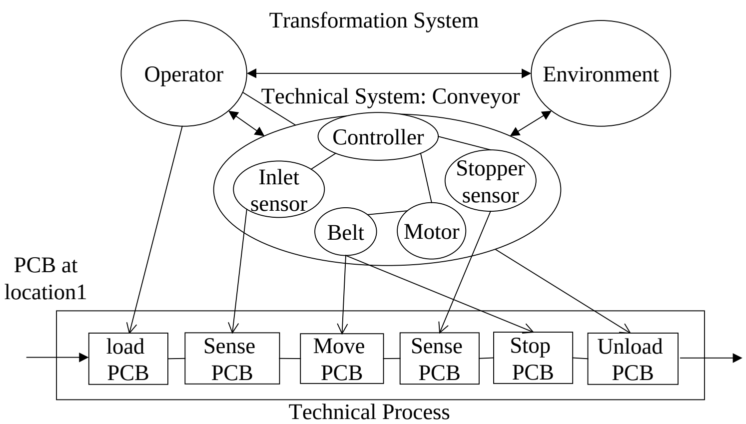


Figure 3 Transformation Model for Conveyor

b) Create/select parts from parts library
   If parts are not available, the user will need to add a new part to the database, otherwise the operator and operand can be selected from the parts library.
c) Create part properties and states
   This step is optional for a part which has already had historical data related to it, otherwise new properties and states for the parts involved need to be added to the database.
d) Create/select functions and their behaviours
   Again functions and the associated behaviours can be selected or created from the list in the database.

e) Create function units
   Function units which make up the new design can be created from the [Part]s and [Generic Function] information.
f) Create assembly
   Usually a new design is an assembly of parts and functions. A new assembly can be created and stored in a [Model] object.
g) Establish precondition and postconditioin relationship between part state and behaviour
   A part will respond based on the information given to precondition and postcondition in the database. The user can define these important relationships based on their knowledge of the part, or based on the failure reports.
h) Continuous process: Data input based on failure reports
   Normally, a failure report will provide information about the failure (behaviour), the cause of failure and its source (operator state) and the effect of failure (operand state). This information is sufficient to build the precondition and postcondition for a given failure condition. Hence, with continuous data input from failure reports, the capability of the system to respond to a given design will increase. In short, the system is continuously "learning" from past failures.

At first glance, there seems to be many steps involved in the generation of an FMEA. However, as the database has been "enriched" with knowledge, the knowledge can be reused, and the optional steps given above may not be needed.

4.2 Forming a New FMEA Case

In the traditional FMEA method, the knowledge of an FMEA is limited by the case being recorded by the user. Using the proposed FMEA model, the knowledge resides in the part, not in the cases. Hence, the system is able to respond to new cases not previously captured by the user. For example in PCB assembly, failure cases may have been recorded for 'motor moves belt' and 'belt moves PCB'. Hence, should the motor fail, then the belt will not move, and consequently the PCB fails to move.

Hence, the information captured in the database would be:

Function Unit 1:	motor moves belt
Operator:	motor
Generic Function:	move
Operand:	belt
Precondition:	State: motor failure – Behaviour: not moving
Postcondition:	Behaviour: not moving – State: belt not moving

Function Unit 2:	belt moves PCB
Operator:	belt
Generic Function:	move
Operand:	PCB
Precondition:	State: belt not moving – Behaviour: not moving
Postcondition:	Behaviour: not moving – State: PCB not moving

If another user created a design with the function unit: "motor moves PCB", which has never been captured from the failure report, the system will search for the operator with the name "motor" with function "move" and retrieve the likely precondition (State: motor failure – Behaviour: not moving). The same process is carried out on operand with the name "PCB" and function "move". In this case, it retrieves the likely postcondtion (Behaviour: not moving - State:

PCB not moving). The combination of this information will result in a new case: “motor fails, PCB not moving”.

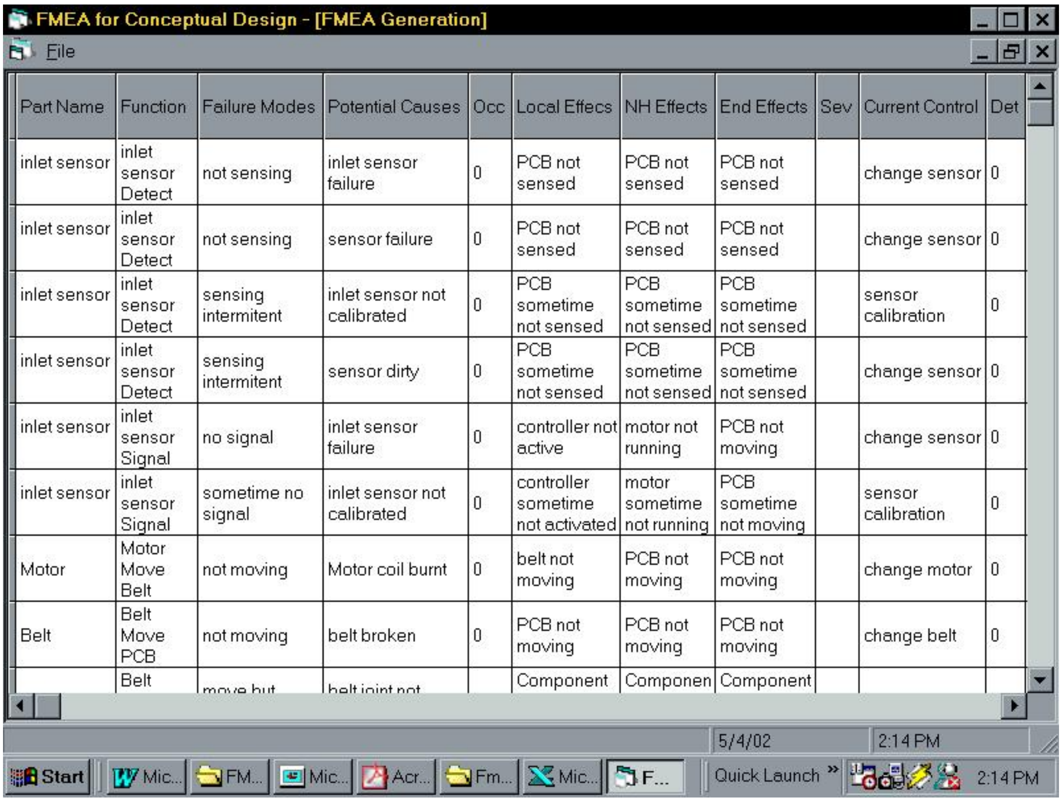
5. CASE STUDY

Failure reports are used as the source for cause and effect chain building which eventually leads to the generation of an FMEA. The failure records of a conveyor and a chip-mounting machine have been used for the case study, courtesy of Motorola Technology Malaysia PLC. Table 1 shows a sample extracted from the failure report for the chip-mounting machine.

Table 1 Sample of Failure Report

ITEM_ID	DATE_ATND	NICKNAME	DATE_RQS	PROBLEM	CAUSE	SOLUTION	DATE_CLOSE
A24517	3/27/02 7:46	SIE-S20-13	3/27/02 7:21	incorrect nozzle lenght	nozzle worn out	replace	3/27/02 8:00
A24531	3/27/02 8:04	SIE-S20-10	3/27/02 8:03	unable to pick-up	-nozzle clogged	-replaced new nozz.	3/27/02 8:06
A24535	3/27/02 8:09	SIE-S20-10	3/27/02 8:09	intermitent communication	-cable loose connection	-fixed properly.	3/27/02 8:17
A24539	3/27/02 8:19	SIE-S20-10	3/27/02 8:19	comp fly/overturn	shutter jammed - shutter bend	change	3/27/02 8:51

All related parts for the conveyor are modelled and captured by the system. As a result, full FMEA generation is created as shown in Figure 4.

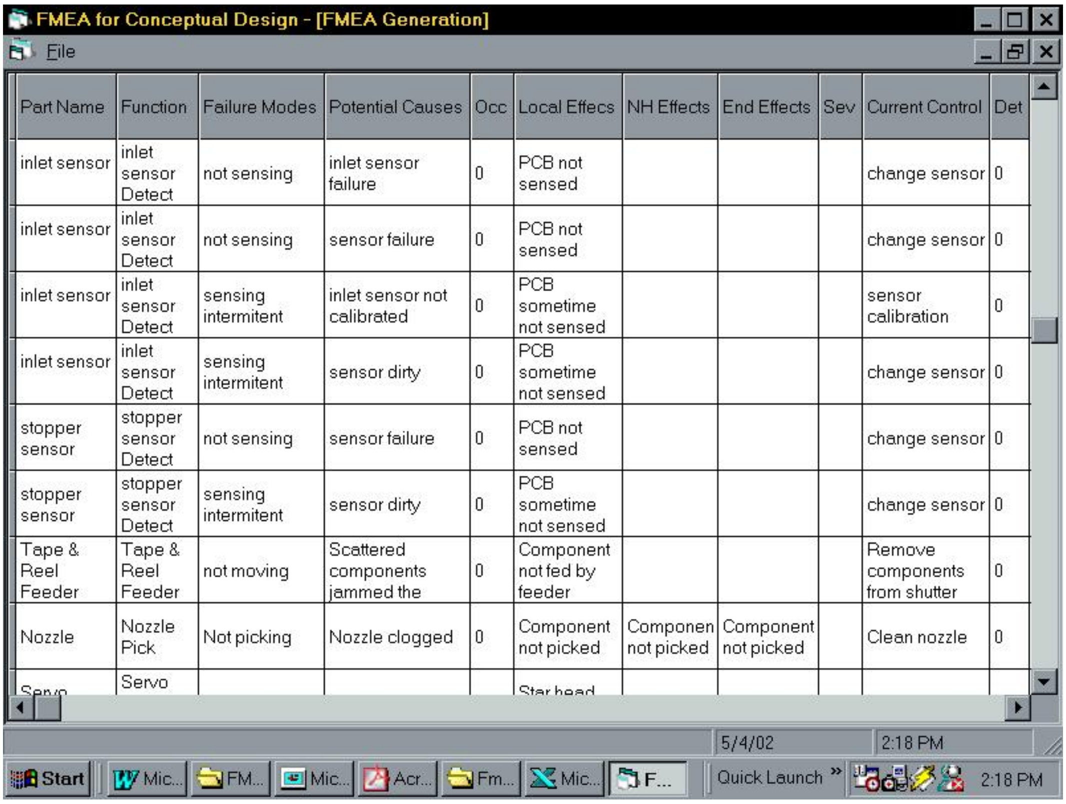


FMEA for Conceptual Design - [FMEA Generation]

Part Name	Function	Failure Modes	Potential Causes	Occ	Local Effecs	NH Effects	End Effects	Sev	Current Control	Det
inlet sensor	inlet sensor Detect	not sensing	inlet sensor failure	0	PCB not sensed	PCB not sensed	PCB not sensed		change sensor	0
inlet sensor	inlet sensor Detect	not sensing	sensor failure	0	PCB not sensed	PCB not sensed	PCB not sensed		change sensor	0
inlet sensor	inlet sensor Detect	sensing intermitent	inlet sensor not calibrated	0	PCB sometime not sensed	PCB sometime not sensed	PCB sometime not sensed		sensor calibration	0
inlet sensor	inlet sensor Detect	sensing intermitent	sensor dirty	0	PCB sometime not sensed	PCB sometime not sensed	PCB sometime not sensed		change sensor	0
inlet sensor	inlet sensor Signal	no signal	inlet sensor failure	0	controller not active	motor not running	PCB not moving		change sensor	0
inlet sensor	inlet sensor Signal	sometime no signal	inlet sensor not calibrated	0	controller sometime not activated	motor sometime not running	PCB sometime not moving		sensor calibration	0
Motor	Motor Move Belt	not moving	Motor coil burnt	0	belt not moving	PCB not moving	PCB not moving		change motor	0
Belt	Belt Move PCB	not moving	belt broken	0	PCB not moving	PCB not moving	PCB not moving		change belt	0
	Belt	move but	belt joint not		Component	Componen	Component			

Figure 4 Generated FMEA for Conveyor

In the case of the chip-mounting machine, not all parts are modelled, i.e. not all parts are provided with the properties, states and behaviours that are required for the FMEA generation. The chip-mounting machine has a conveyor which is similar to a conveyor used in a previous case. Hence, the data can be reused without creating additional parts for its conveyor. The result is as shown in Figure 5. Even if the machine is not fully modelled, it is capable of providing a generated result based on the historical data and user input from the limited failure reports. The user can then complete the FMEA manually.



Part Name	Function	Failure Modes	Potential Causes	Occ	Local Effecs	NH Effects	End Effects	Sev	Current Control	Det
inlet sensor	inlet sensor Detect	not sensing	inlet sensor failure	0	PCB not sensed				change sensor	0
inlet sensor	inlet sensor Detect	not sensing	sensor failure	0	PCB not sensed				change sensor	0
inlet sensor	inlet sensor Detect	sensing intermitent	inlet sensor not calibrated	0	PCB sometime not sensed				sensor calibration	0
inlet sensor	inlet sensor Detect	sensing intermitent	sensor dirty	0	PCB sometime not sensed				change sensor	0
stopper sensor	stopper sensor Detect	not sensing	sensor failure	0	PCB not sensed				change sensor	0
stopper sensor	stopper sensor Detect	sensing intermitent	sensor dirty	0	PCB sometime not sensed				change sensor	0
Tape & Reel Feeder	Tape & Reel Feeder	not moving	Scattered components jammed the	0	Component not fed by feeder				Remove components from shutter	0
Nozzle	Nozzle Pick	Not picking	Nozzle clogged	0	Component not picked	Componen not picked	Component not picked		Clean nozzle	0
Servo	Servo				Star head					

Figure 5 Generated FMEA for Chip-Mounting Machine

The result of the FMEA generation depends very much on the data input from the user. An inaccurate input may cause the system to provide a false result. A precautionary step is taken so that the user has the final decision on whether to accept the result of an FMEA generation, and save it into the FMEA file. The user can always go to the software to rectify the input later.

6. CONCLUSION

This paper has described current shortcomings faced by FMEA users, and proposed a way for FMEA generation to overcome those difficulties. The proposed model is based on the study of various methods applied in FMEA research. The model is intended to be more generic in its application so that it can be applied to many design cases including product and process design. The proposed system successfully demonstrates FMEA generation based on user input, and the reuse of existing knowledge. Only two manufacturing process cases have been used so far. The ongoing developments include:

- Extending the cases study to include more cases including design cases for verification
- Enabling a part to inherit the characteristics of another part which is considered as its "parent" in the parts library
- Application to multiple users

ACKNOWLEDGEMENTS

The authours wish to express their gratitude to the Mechanical and Manufacturing Department of Loughborough University and Motorola Technology Malaysia for providing support for this research.

REFERENCES

[1] R.I.M. Young, O. Canciglieri-Jnr, C.A. Costa, Information Interactions for Data Model Driven Design for Manufacture, Globalization of Manufacturing in the Digital Communications era of the 21st Century: Innovation, Agility and the Virtual Enterprise, Kluwer Academic Publishers, 1998, pp.313-324.

[2] Tien-Chien Chang, Richard A. Wysk, Hsu-Pin Wang, Computer-Aided Manufacturing, 2nd Edition. Prentice Hall International Series in Industrial and System Engineering, 1998.

[3] BS 5760 Part 5, 1991: Reliability of systems, equipment and components. Guide to failure modes, effects and criticality analysis (FMEA and FMECA)

[4] B.G. Dale, P. Shaw, Failure Mode and Effects Analysis in the U.K. Motor Industry: A state-of-the-art Study, Quality & Reliability International, 1996

[5] Christopher J. Price, David R. Pugh, Myra S. Wilson, Neal Snooke, Flame system: automating electrical failure mode & effects analysis (FMEA**)**, Proceedings of the Annual Reliability and Maintainability Symposium, (1995), p 90-95

[6] Wynne Hsu, Irene M. Y. Woon, Current research in the conceptual design of mechanical products, Singapore, Computer Aided Design, v 30, n 5, (Apr 1998), p 377-389 ISSN: 0010-4485 Publisher: Elsevier Sci Ltd

[7] Rudiger Wirth, Bernd Berthold, Anita Kramer, Gerhard Peter. Knowledge-based Support of System Analysis for Analysis of Failure Modes and Effects, Engineering Applied Artificial Intelligent, Vol. 9, No.9, pp. 219-229, 1996, Elsevier Science Ltd.

[8] B. H. Lee, Design FMEA for Mechatronic Systems Using Bayesian Network Causal Models, Proceedings of the ASME Design Engineering Technical Conferences, ISBN: 079181971X, 1999, vol. 1, pp.1235-1246.

[9] J. E. Hunt, D. R. Pugh, C. J. Price, Failure mode effects analysis: a practical application of functional modelling, Applied Artificial Intelligence, v 9, 1, (Jan-Feb 1995), p 33-44

[10] Charles F. Eubanks, Steven Kmenta, Kosuke Ishii, Advanced Failure Modes and Effects Analysis Using Behavior Modelling. Proceeding of DETC'97, 1997 ASME Design Engineering Technical Conference and Design Theory and Methodology Conference, September 14-17, 1997, Sacramento, California.

[11] J.S. Gero, K.W. Tham, H.S. Lee, Behaviour: A Link Between Functional and Structure in Design. Proceeding of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design, Elsevier Science Publisher, 1991

[12] D.J. Russomanno, RD. Bonnell, J.B. Bowles, Functional reasoning in a failure modes and effects analysis (FMEA) expert-system. Proceedings of the Annual Reliability and Maintenability Symposioum, 1993, pp. 339-347

[13] Vladimir Hubka, W. E. Eder, Theory of Technical Systems: A Total Concept Theory for Engineering Design, Springer-Verlag Berlin, 1984.

[14] R.M. Atkinson, M.R. Montakhab, K.D.A. Pillay, D.J. Woollons, P.A. Hogan, C.R. Burrows, K.A. Edge, Automated Fault Anlaysis for Hydraulic Systems. Part 1: Fundamentals. Proceeding of the Institution of Mechanical Engineers, Part 1, Journal of Systems and Control Engineering, vol. 206 Part 14, 1992, pp. 207-214

[15] N. Hughes, E. Chou, C. J. Price, M. Lee, Automating mechanical FMEA using functional models. Proceedings of the Twelfth International Florida AI Research Society Conference. AAAI Press. 1999, pp. 394-8