Modelling and reasoning for failure modes and effects analysis generation

P C Teoh and **K Case***
Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire, UK

Abstract: Failure modes and effects analysis (FMEA) is a quality improvement and risk assessment tool commonly used in industry. It is a living document used to capture design and process failure information. However, the traditional FMEA has its limitations in terms of knowledge capture and reuse. In order to increase its effectiveness, much research has been carried out to find an effective way to provide FMEA generation. However, because of the complexity of the information needed, most of the research concentrates on the application for a specific design domain. This paper reviews various FMEA research studies and modelling and reasoning methods that can be used for generic applications. A new proposal made is based on the 'knowledge fragment' reasoning concept suggested by Kato, Shirakawa and Hori in 2002. FMEA is introduced in the conceptual design stage so as to minimize the risks of costly failure. The method enables new knowledge to be formed using the limited available information in the conceptual design stage. A prototype has been created to evaluate the proposed method. Case studies have been conducted to validate the proposed method. The case studies show that the method is able to provide reliable results with limited information.

Keywords: failure modes and effects analysis (FMEA), conceptual design, modelling, causal reasoning, functional model

1 INTRODUCTION

Concurrent engineering is an initiative to improve the competitiveness of manufacturing industry. The general aims are to improve quality, to reduce cost and to reduce cycle times of the products. Many tools have emerged in line with this initiative. One, which has been adopted by the International Standard Organization, is failure modes and effects analysis (FMEA) [**1**].

FMEA is a tool used to identify the potential failure modes of a product or a manufacturing process, and the effects of the failures, and to assess the criticality of these effects on the product functionality. It provides basic information for risk assessment and quality improvement of product and process design. According to BS 5760: Part 5 [**2**], '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.' When the criticalities of the failures are assessed, the method is known as FMECA. Hence FMECA is an extension of FMEA. In this research, FMEA and FMECA are treated as the same method. The method will include critical analysis and be known only as FMEA.

Basically, FMEA can be classified into two main types, i.e. design FMEA and process FMEA. Design FMEA deals with product design, while process FMEA is used to solve problems due to manufacturing processes. The potential failure modes and potential causes for each component or process step are identified. This is followed by assessment of the failure effects to the end users. The risk of each failure is prioritized on the basis of the risk priority number (RPN). RPN is a decision factor based on the product of three ratings: occurrence, severity and detection. These ratings are scaled with numbers between 1 and 10. Failure modes with high RPN values are selected. The corresponding current controls (i.e. the solutions) will be implemented on the basis of the selected failures.

2 MOTIVATION

Traditionally, potential problems of a design or process are captured with FMEA manually using hard copy or

The MS was received on 5 June 2003 and was accepted after revision for publication on 28 November 2003.
**Corresponding author: Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK.*

spreadsheet. However, as the accumulated FMEA knowledge grows, the information becomes increasingly difficult to find. Hence, it is increasingly harder to reuse.

It is very difficult to implement a highly manual FMEA (i.e. a report that is keyed in manually on to paper or into a spread sheet). The manual method is found to be not user friendly, hard to understand and of limited flexibility. As a result, many companies use FMEA merely to satisfy the contractual requirements of their customers [**3**]. Users may find FMEA a 'tedious and time-consuming activity' [**4**]. FMEA is often carried late in the design cycle after the design prototype has been built [**4**], and the changes made at later stages will be very costly. Hence, there is considerable research that attempts to improve FMEA usage in the earlier stages of the design process, such as the conceptual design stage.

Much research has been carried out mainly to provide automatic FMEA report generation. The research reviewed in this paper includes FLAME [**4**] and Auto-Steve^{TM} [**5**] for the design of automobile electrical systems, GENMech [**6**] for mechanical design and research by Atkinson *et al.* [**7**] and Hogan *et al.* [**8**] for hydraulic systems design. Bouti *et al.* [**9**] and Price *et al.* [**10**] suggested methods for process FMEA application. Eubanks *et al.* [**11, 12**] proposed a more generic approach for both design and process FMEA. However, most of the methods require a considerable amount of modelling effort to be used effectively. Hence, despite all the efforts, most of the mechanical, electromechanical and manufacturing process designs still use the conventional method to create an FMEA.

In order to improve FMEA usage in the early design stages, artificial intelligence (AI) techniques such as modelling and reasoning are used. This paper specifically looks at a modelling and reasoning approach that provides the basis for FMEA automation for more generic product and process design applications.

3 MODELLING AND REASONING

Modelling and reasoning are two important and widely used concepts in FMEA research. A model is an abstracted picture of a concept. A model may represent a system, an object or a problem constructed for the purpose of analysis [**13**]. It is an approximation of the real thing. Modelling is a process of transferring the concept into a type of representation that people can comprehend, communicate and work upon.

Reasoning is a decision-making process based on the understanding of the available information. In AI terms, reasoning represents the capability of the computer to make decisions based on the given information.

These two concepts are dependent on each other in executing a task. In FMEA, a model can be used to represent a product or the component of a product (the structure), as well as the design intent (the function) of the product. A reasoning technique defines the causal relationships between the information of the structures and functions in the model.

3.1 Modelling in FMEA research

The models used in FMEA research can be divided into two types, i.e. functional models and structural models. Both types of model are needed to automate the FMEA process [**14**].

A functional model describes the intended function or the purpose of a system. The functional model is made up of two main components: function and behaviour. The function of a system provides the design intent, whereas the behaviour describes how the structure of an artefact achieves its function [**15**]. A function can be decomposed into subfunctions to understand better the design through functional analysis. This will be further discussed in a subsequent section.

A structural model is defined as 'the components that make up an artifact and their relationships' [**15**]. It refers to the configuration of the product or system. It contains the information of all the components, entities, subprocesses or subsystems, and the interactions between them that make up a useful structure for an intended purpose. A structural model may refer to a physical assembly of a mechanical or electrical product (such as a car, an engine or an electrical circuit), or a software configuration.

In design, each artefact is created to achieve one or more functions. At the same time, one or more artefacts can achieve a function. The relationships between functions and artefacts are represented by the mapping between a functional and a structural model (Fig. 1), as defined by Eubanks *et al.* [**12**].

3.2 Modelling in conceptual design

There is much literature that suggests a systematic design approach for the design process [**16–18**]. In general, the design process can be divided into four phases:

1. *Design specification*. Establish the requirement specifications.
2. *Conceptual design*. Find the possible design concepts based on the requirements.
3. *Embodiment design*. Design the layout, schematic, draft or configuration drawing of the design.
4. *Detail design*. Establish the detail dimensions using proper engineering drawings.

Conceptual design is a phase where ideas are generated, evaluated and selected. The outputs of this phase are the design concepts that will be the basis for the embodiment and detail design. Briefly, the steps include formulating the problem, establishing a functional

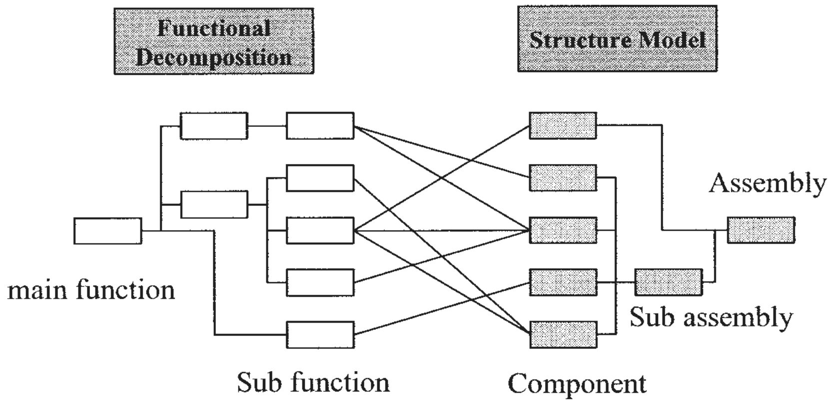

Fig. 1 Function–structure mapping [12]

model, searching for working principles or function carriers, combining suitable working principles into one or more concept variants and evaluating the concepts. One or more selected concepts will go through embodiment design, where the layout and forms of the design will take shape. Further evaluations may take place in that phase.

The functional model consists of the decomposed functions of the design used to simplify the design problem in order to search for suitable working principles. Hence, function decomposition provides the first step for FMEA involvement in the design process. The overall process involved in function decomposition and searching for working principles is known as functional analysis.

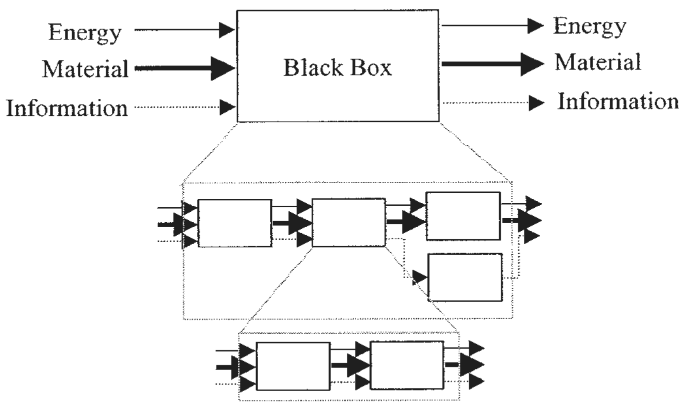

Fig. 2 Functional model with energy, material and information flows

4 FUNCTIONAL ANALYSIS

There are many ways to achieve function decomposition, including the flow-based approach, the integrated definition (IDEF) method and a functional diagram approach.

4.1 Flow-based black-box approach

Pahl and Beitz [17] and Ulrich and Eppinger [18] suggested a flow-based 'black-box' design approach. The black box represents the function of a design or process. Normally, a function is represented by a verb–noun pair [e.g. 'moves printed-circuit board (PCB)']. The role of the function is to convert an input into an output of a different state. The inputs and outputs of the black box (the operands) can be represented by three basic elements, i.e. energy, material and information (or signal) flows. The main function in the functional model can be decomposed into many subfunctions, forming one or more alternative functional structures (Fig. 2).

The main issue in using a flow-based approach in conceptual design is that the operand needs to be in the basic form of the three types of flow. For example, if a motor carries out the function 'move pulley' this can only be represented by further decomposition into the following: motor 'generates torque', and torque 'rotates pulley'. In contrast, using an implicit concept, motor 'moves pulley' implies the actions 'generate torque' and 'rotate pulley'. The flexibility in modelling implicit concepts will release the designer from some burdens so that they can concentrate on design solutions. Ulrich and Eppinger [18] also pointed out that, in some applications, the energy, material and information flows are difficult to identify. Hence, a flow-based approach imposes a restriction that hinders the formation of implicit relationships in a model.

4.2 IDEF methods

Kusiak [13] showed that a process model could be represented by IDEF methods. IDEF methods are standard methodologies that are widely used in concurrent engineering. IDEF represents a family of modelling methods including IDEF0 for function models, IDEF3 for process models, etc.

The IDEF0 diagram introduced by the National Institute of Standards and Technology [19] is shown in Fig. 3. The function name is a defined verb or verb phrase. The input and output arrows represent the operand of the system. A control from the top represents

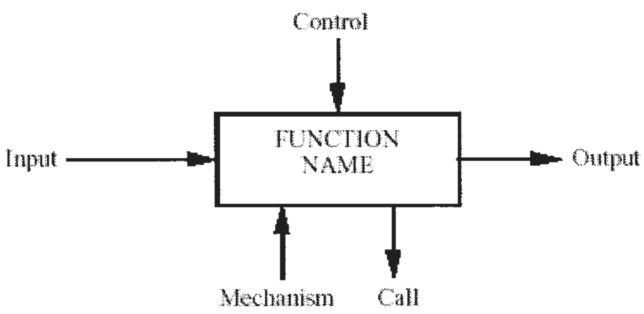

Fig. 3 IDEF0 function box

the elements that influence the function performance. A mechanism is a means that enables the function to be performed. A call arrow represents the communication from one function box to another. A function box can be decomposed into more detailed subfunctions, similar to the flow-based 'black-box' approach.

According to Dorador [**20**], IDEF0 lacks the capability to represent processes involving time and sequence (precedence relationships). Hence, the IDEF3 diagram is introduced to provide process modelling capability. The IDEF3 diagram is provided with logical connectors to represent timing and sequences, as shown in Fig. 4.

According to Mayer *et al.* [**21**], the IDEF0 diagram can be used at the initial stage of complex model building where precedence relationships are not clear. Decomposition of the initial model will lead to a level where the IDEF3 diagram is used to represent process models. Kusiak [**13**] maintained the control and mechanism arrows from IDEF0 in IDEF3 diagrams. This is very helpful in representing the structure information in terms of 'mechanism' and 'control' of the function.

The downside of using IDEF methods is that the method is not suitable for static model functions, i.e. a function that is achieved due to the structure of the operand and not due to the state-change behaviour [**22**]. For example, a function 'support' will only maintain the state of the operand. In fact there is no sequence or direction involved that can be used to construct an IDEF diagram.

4.3 Functional diagram

The functional diagram provides one of the simplest models to represent function and structure interaction. The basic unit (function unit) of the diagram consists of two objects linked by a function. The first object is the component in the structure model that acts as the operator to the function. The function is a verb or verb phrase that defines the action. The second object is the operand of the model. An operator in one function unit can be an operand of the other function unit and vice versa. Hence, the objects and functions are interconnected to form a network known as the functional diagram (Fig. 5).

The attractive aspects of applying functional models in conceptual design are their simplicity and user-friendliness. This is very important for conceptual design since frequent design changes are essential in this phase.

However, because of its simplicity, the method lacks other features in modelling. It cannot handle timing and sequences as in IDEF3. The diagram can be too complicated when it is used to represent a complex design. There is no standard terminology for defining the objects and the function, and hence it is not efficient in reuse.

The modelling techniques reviewed above have their own advantages and weaknesses for conceptual design applications. In this research, a combination of modelling techniques can be used. The functional diagram is used for conceptual modelling, as it is the simplest. Its weaknesses can be complemented by other modelling techniques. The IDEF method is the common method used in the industry. Hence, it provides a launch pad for conceptual design. For example, IDEF3 diagrams

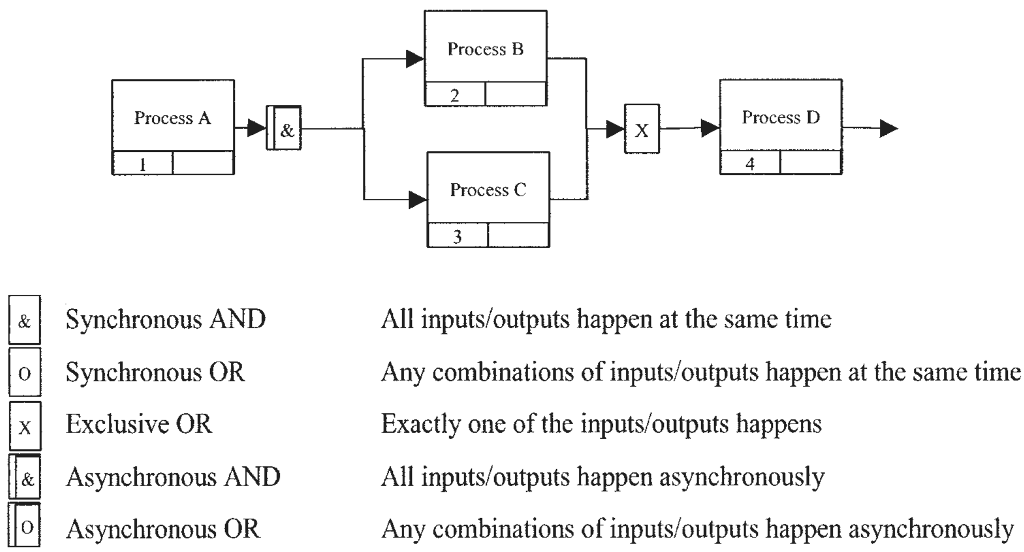

Fig. 4 IDEF3 diagram with logical connectors

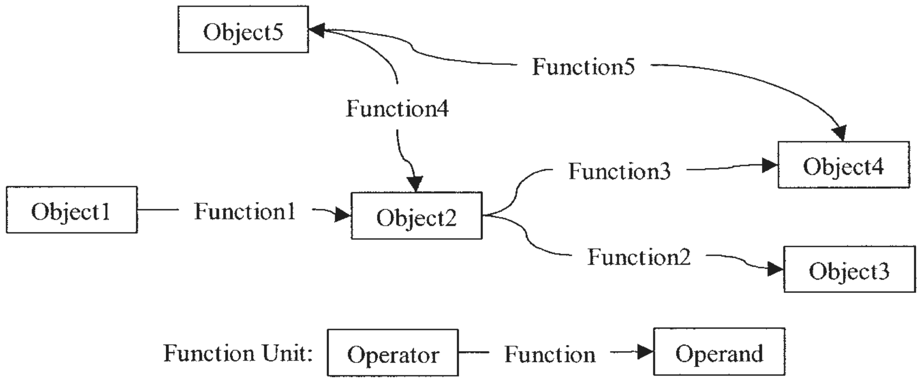

Fig. 5 Functional diagram

can be used to build the initial model before transforming to the functional diagram. The method will be discussed in the following section.

5 REASONING IN FMEA

Reasoning in a design process is to search for possible function and structure mapping. In FMEA, reasoning is carried out to establish cause-and-effect relationships based on the functional and structural models, and to generate the FMEA report.

5.1 Common reasoning approaches

There are three common reasoning approaches in AI, i.e. rule-based reasoning, model-based reasoning and case-based reasoning. According to Maher *et al.* [**23**], rule-based reasoning uses IF–THEN rules to capture the knowledge. Domain experts are usually needed to identify these rules. Model-based reasoning aims at formulating knowledge in the form of principles. These principles are more general than the IF–THEN rules. Hence this method is applicable to a wider range of problems than the rule-based method. Case-based reasoning is an experience-based method that associates prior problem experiences with the current cases. Thus specific cases and the corresponding prior experience form the main knowledge sources for a case-based reasoning system.

There are different reasoning approaches suggested in FMEA research. The model proposed by Bouti *et al.* [**9**] is rule based. The rules are built within the functional blocks to reason about the failure modes, causes and effects. Since a shallow knowledge reasoning approach has been used, the rules are generic for all function blocks. Hence, it avoids complexity due to custom-made rules for different functions. The disadvantage of this approach is that it relies on the data input to the system.

Model-based reasoning has been used in some methods, such as those of Price [**5**], Hughes *et al.* [**6**], Atkinson *et al.* [**7**] and Hogan *et al.* [**8**]. In actual fact, there are rules residing in many of the model-based systems. A model-based approach can provide an accurate simulation of the failure conditions. However, a rather comprehensive structural model needs to be created before the reasoning process can be carried out. A component library for the predefined models needs to be created to eliminate modelling activities during FMEA generation. These projects focus on specific areas such as electrical circuits for cars [**5**], or hydraulic components [**7**, **8**]. Owing to the complexity of the manufacturing process, to use this approach alone in process FMEA will require a complex structure (component) model, which may not be practical.

Case-based reasoning has been used in problem diagnosis [**10**] but not for FMEA generation. Case-based reasoning relies on historical cases. The information of the conditions during the occurrence of a failure can constitute a case for the reasoning process. Hence, it is theoretically possible to apply case-based reasoning in FMEA generation. However, the information supplied to the cases must be comprehensive enough to provide an accurate result. This could be a problem for conceptual design, where most of the information is still lacking.

5.2 'Knowledge fragment' reasoning approach

Besides FMEA research, reasoning has been applied to conceptual design and problem solving. Kato *et al.* [**24**] suggested a 'knowledge fragment' approach for reasoning in a problem-solving tool. Previous failure reports (fault cases) are knowledge fragments that reflect the deliberation, reasoning and experience of the experts. Each knowledge fragment is highly reusable. Initially, a model (Fig. 6) must be constructed using function and component ontologies. The failure reports can be represented by the schema shown in Table 1. Assuming that there were previous failure reports recorded using the schema, when a user provides a failure mode to one of the components in the functional model, the tool will compute all possible paths based on the functional links among the components in the model.

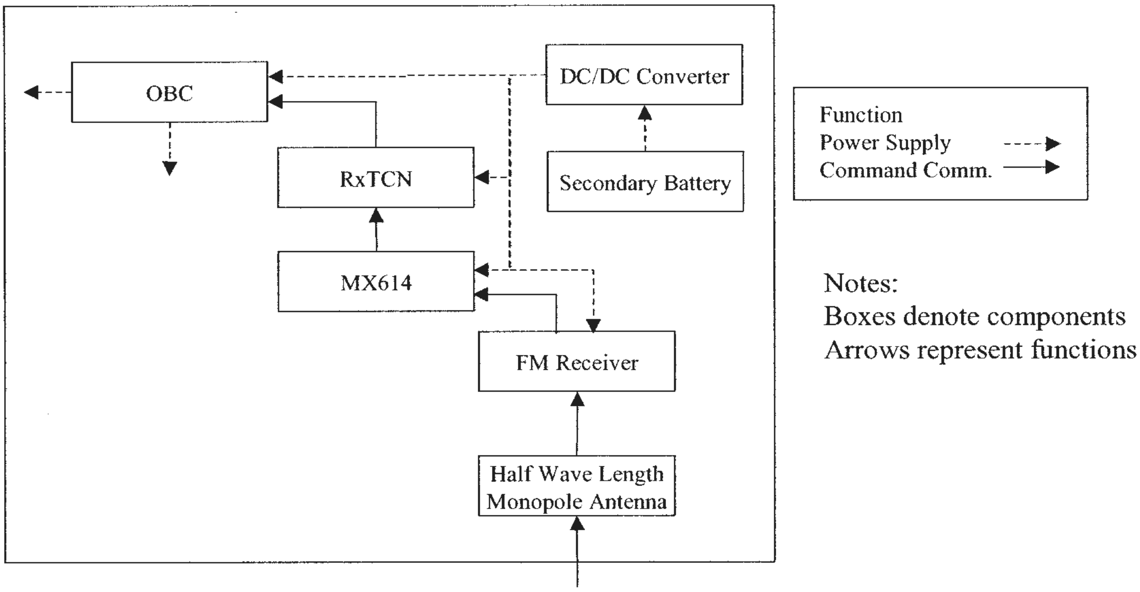

Fig. 6 Part of the satellite functional model [**24**]

Table 1 Schema for failure report [**24**]

Item	Description
Label	The label of the fault case
Affected component	The component that failed in the fault case
Affected function	The function of the affected component that was impaired in the fault case
Cause function	The function of the cause component on which the affected component depended to be operational
Details	The description of details of the fault case

Using the satellite example given by Kato *et al.* [**24**], the voltage of the secondary battery in Fig. 6 is affected by low temperatures. Hence, the cause function 'temperature dependence' has influenced the affected function 'power supply' (second column in Table 2). In a d.c.–d.c. converter, the affected function 'power supply' from the secondary battery has become the cause function to the d.c.–d.c. converter. Hence the failure has been propagated to the d.c.–d.c. converter (third column in Table 2). This will eventually lead to the frequency-modulated receiver, where the function 'command reception' will be affected (fourth column in Table 2).

The advantage of this approach is that reasoning can be carried out on the basis of a relatively small amount of information. Models are driven by information assigned to the ontologies rather than basic principles and can be easily composed from simple heuristic rules using shallow knowledge reasoning. Hence, it is a suitable method for reasoning in conceptual design.

6 PROPOSED METHOD

Using a combined method based on the above review, a new method known as FMAG (for FMEA generation) was created to automate the generic FMEA report generation. It can be illustrated by the following example.

The IDEF3 diagram in Fig. 7 can represent a process where a PCB is conveyed through a conveyor. The diagram contains information about the functions and operands that can be mapped to relevant structures through functional units. The function units can be

Table 2 Failure case example

Label	Temperature dependence of secondary battery	Inability of d.c.–d.c. converter to boost up voltage	Inability of receiver to receive commands
Affected component	Secondary battery	D.c.–d.c. converter	Frequency-modulated receiver
Affected function	Power supply	Power supply	Command reception
Cause function	Temperature dependency	Power supply	Power supply
Details	When the temperature of a battery becomes low, the voltage provided by the battery becomes low	When the input voltage becomes low, the d.c.–d.c. converter cannot provide enough boost up to the output voltage	There were some cases where commands could not be received when the input voltage to the frequency-modulated receiver becomes low

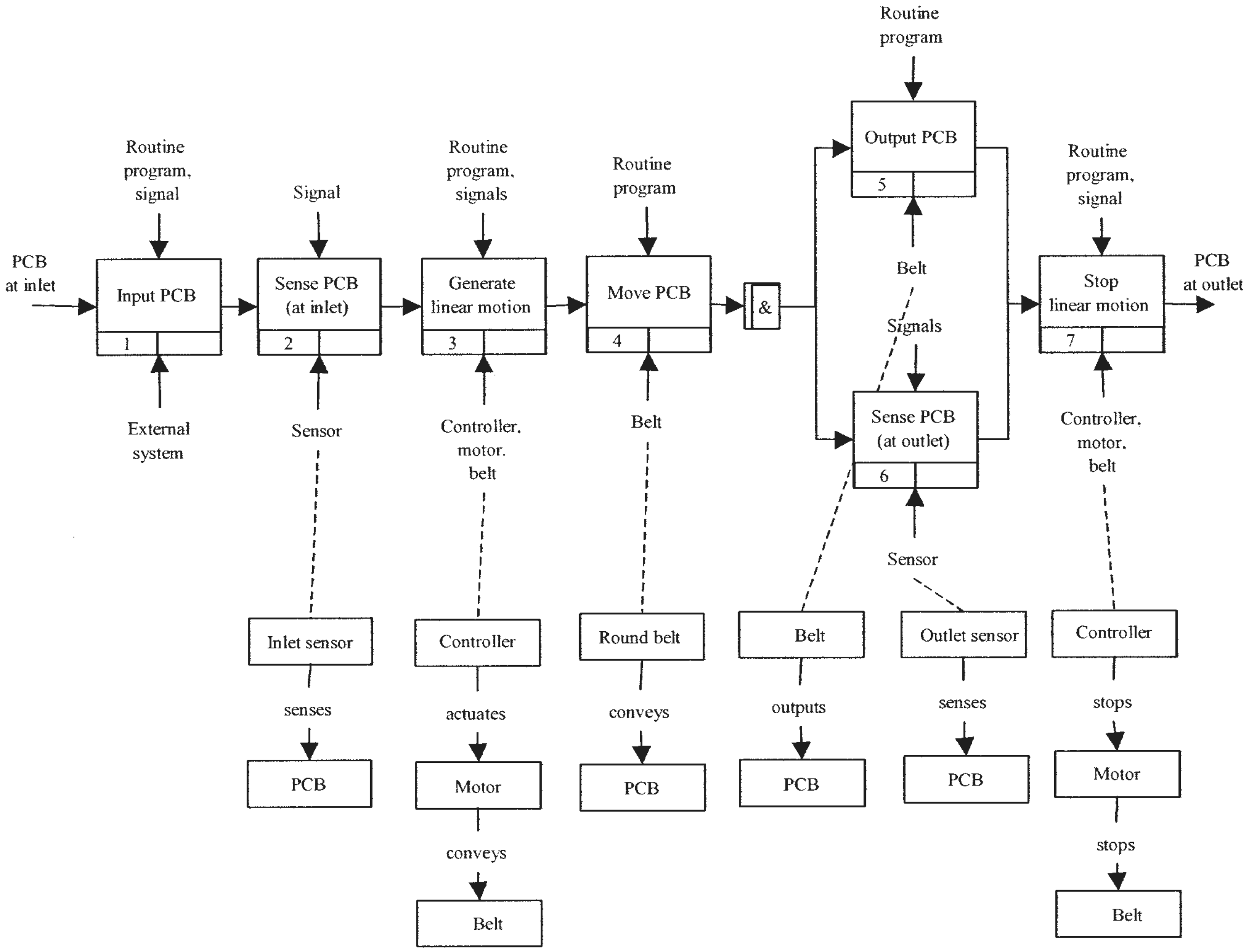

Fig. 7 Function and structure mapping

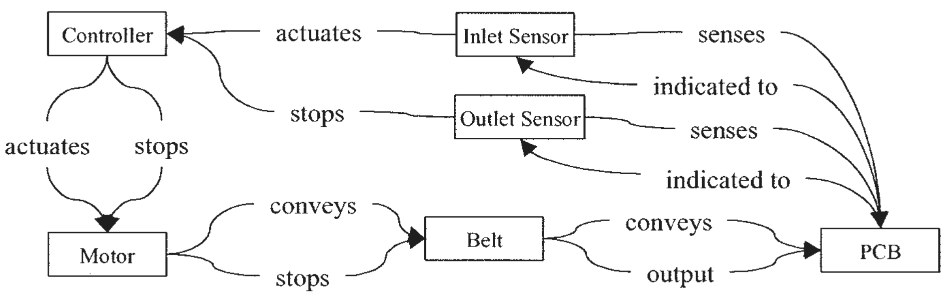

Fig. 8 Functional diagram

combined to form a functional diagram, as shown in Fig. 8.

6.1 Cause-and-effect propagation

In order to facilitate cause-and-effect propagation, a functional diagram must be able to respond to stimulation or changes of state in its components. The causal reasoning drives this response.

As discussed in the previous section, the 'knowledge fragment' reasoning approach [**24**] has been employed. However, unlike the work of Kato *et al.* [**24**] where both cause knowledge and effect knowledge were stored under the same schema, FMAG divides the knowledge fragment into two parts. They are stored separately in 'precondition' and 'postcondition' in the forms of 'operator failure state–failure behaviour' and 'failure behaviour–operand failure state'.

The causal reasoning in FMAG is based on two basic assumptions:

1. There exists a state of an operator where, if there is a change to that state, it will cause its functional behaviour to change accordingly.
2. There exists a functional behaviour where, if there is a change to that behaviour, it will cause the correspond operand to change its state accordingly.

The semantics of the knowledge fragments for the precondition are based on assumption 1, whereas that for the postcondition is based on assumption 2.

The precondition and postcondition gain knowledge fragments through historical data extracted from failure reports and the FMEA. For a particular function unit, the operator state and the behaviour of a failure event form a set of preconditions. The behaviour and the state of the operand form the postcondition of the same event. Hence, with the accumulated events being recorded, precondition and postcondition tables (Figs 3 and 4) will be formed.

Unlike other reasoning approaches, only the minimum information is used here. The reasoning process is carried during the conceptual stage where much information is still not available. During a reasoning process, only the failure cause and effect will be required from the precondition and postcondition tables. The other properties of the operator and operand that remain in normal conditions are assumed and will not be required in the model.

A cause-and-effect propagation method can be used to simulate the actual behaviour of a design in the real world. In a functional model, a state change in one entity will affect the status of the interrelated entities. For example, a PCB that moves into the sensing range of an inlet sensor will provide a good target for the sensor. The sensor will have a state change from 'not sensing' to 'sensing'. This state change will trigger a change in the controller from 'not active' to 'activated'. The controller will in turn enable a motor. The movement of a PCB is the cause that triggers changes across the components in the model. The model is said to have a cause-and-effect propagation.

The propagation is carried out through the behaviour of a generic function via the precondition and postcondition. 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 postcondition relationship. For example, an inlet sensor senses a PCB. The operator is 'inlet sensor', the generic function is 'senses' and the operand is 'PCB'. If a state description 'sensor failure' is introduced into the operator 'inlet sensor', the behaviour for the function 'senses' is 'not sensing', and the operand state is 'PCB not sensed'. Hence, the precondition relationship is 'sensor failure–not sensing', and the postcondition relationship is 'not sensing–PCB not sensed'. Figure 9 shows the precondition–postcondition relationship in a [function unit].

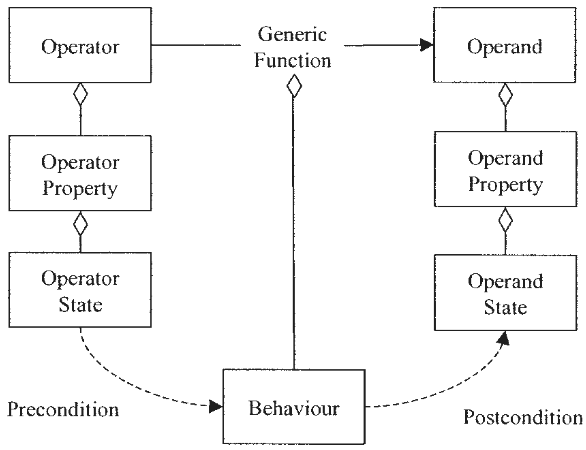

Fig. 9 Precondition–postcondition relationship

The knowledge is referred to the entities and their functions, but not to the function units. During the reasoning process, it is possible to create new knowledge by matching the precondition knowledge and postcondition knowledge with the same failure behaviour. This is demonstrated by the following example.

In a conveying process, the function of the motor is to move the conveyor belt. The belt in turn is intended to move the PCB laid on top of the belt. At an event when the motor fails, the belt will not move, nor does the PCB. Hence the knowledge captured in precondition and postcondition tables can be arranged as in Tables 3 and 4.

The precondition table defines the behaviour of the motor when it fails, and the behaviour of the belt when it is not moving. The postcondition table provides knowledge about the response of the belt when it receives the behaviour 'not conveying' from an operator that is supposed to make the belt move. The postcondition table also provides knowledge about the response of the PCB when it receives the behaviour 'not conveying'. The knowledge is resident in the entities motor, belt and PCB, and not in the function units. This

Table 3 Precondition table

Operator	Generic function	Precondition
Motor	Conveys	Motor failure–not conveying
Belt	Conveys	Belt not moving–not conveying

Table 4 Postcondition table

Generic function	Operand	Postcondition
Conveys	Belt	Not conveying–belt not moving
Conveys	PCB	Not conveying–PCB not moving

approach provides modularity for the creation of new knowledge.

When a new function unit is used in a functional diagram, the operator, operand and the generic function involved can be used as keys to search for the matching states and behaviours in the precondition and postcondition tables. Hence, an entity is able to act or respond to the system through its historical knowledge.

Generating the same result with an identical function unit is straightforward. However, there is a possibility that new knowledge can be generated using a new function unit. Using the same precondition and postcondition tables as above, consider the situation where another designer is creating a design with the new function unit 'motor conveys PCB'. Assuming that the function unit has never been captured from the failure report, under normal circumstances, the knowledge will not be available for reasoning. However, FMAG provides a means to create new knowledge based on possible matching between information in the precondition and postcondition tables.

The system will search for the operator with the name 'motor' with function 'conveys' and retrieve the likely precondition 'motor failure–not conveying'. The same process is carried out on the operand with the name 'PCB' and function 'conveys'. In this case, it retrieves the likely postcondition 'not conveying–PCB not moving'. The combination of this information will result in a new case 'motor failure–PCB not moving'. Hence, PCB has the knowledge to respond to the motor failure even though the case has never existed.

The failure states of an object are not limited to binary states ('move', 'not moving', etc.) but can also include states with intermittent failures such as 'sometimes not moving' or 'intermittent movement'.

6.2 FMEA generation

FMEA generation is achieved when the causal reasoning technique is applied throughout the functional diagram. When a new functional diagram is created for a particular design, the FMEA report is generated on the basis of historical data saved in the database. The user can provide additional information such as the RPNs, current control and recommended action at certain stages of the FMEA generation process. Table 5 shows an example of the generated FMEA.

6.3 Hierarchical functional modelling

The current FMAG is limited to provide cause-and-effect propagation within a single level of abstraction; i.e. state changes of all the objects can only be represented within one functional diagram. Further developments could be made so that different functional diagrams are used to represent the design models in different levels of abstraction.

An abstract model can be decomposed into more detailed submodels so that analysis can be carried out, and this decomposition can be carried out at many levels of abstraction. In terms of FMEA application, multiple-level modelling enables analysis to be carried out across different levels of abstraction. In fact, BS 5760: Part 5 [**2**] discusses effects propagation in FMEA for both single and multiple levels.

Under the current FMAG structure, the function units are connected to form a functional diagram. Each functional diagram represents a scenario of a design operation or process. A model is described by different scenarios of the design or process. A model is a part of the entity class; hence it can serve as an operator or an operand of yet another functional diagram. The decomposed model can be represented by the example in Fig. 10.

At the lowest level, generic functions in the functional basis are used in the functional models. However, at higher levels, non-standard terms are used. As shown in Fig. 10, many of the function units in the second level functional diagram use the function 'produces', which is not a generic function. The objects and functions in the functional diagram at the lower level are aggregations of the higher-level model. Hence, the hierarchical functional diagrams represent the decomposition of structural and functional models, and the relationships of the entities in both models.

7 CASE STUDIES

Prototype software has been created on the basis of the FMAG method. Two design cases and three process cases for two-way radio design and manufacture have been evaluated using the prototype software. Information from previous FMEA reports and the failure reports have been entered into the software. Figure 11 shows a screen dump of the prototype software.

Table 5 An example of a generated FMEA item

Part/ process step	Part/process step functions	Potential failure modes	Potential causes	Occurrence	Local effect	Next high-level effect	End effect	Seventy	Current controls	Detection	RPN
Inductive motor	Conveys round belt	Not conveying	Carbon brush wear and tear	3	Belt not moving	PCB not moving	PCB not moving	4	Change carbon brush	4	48

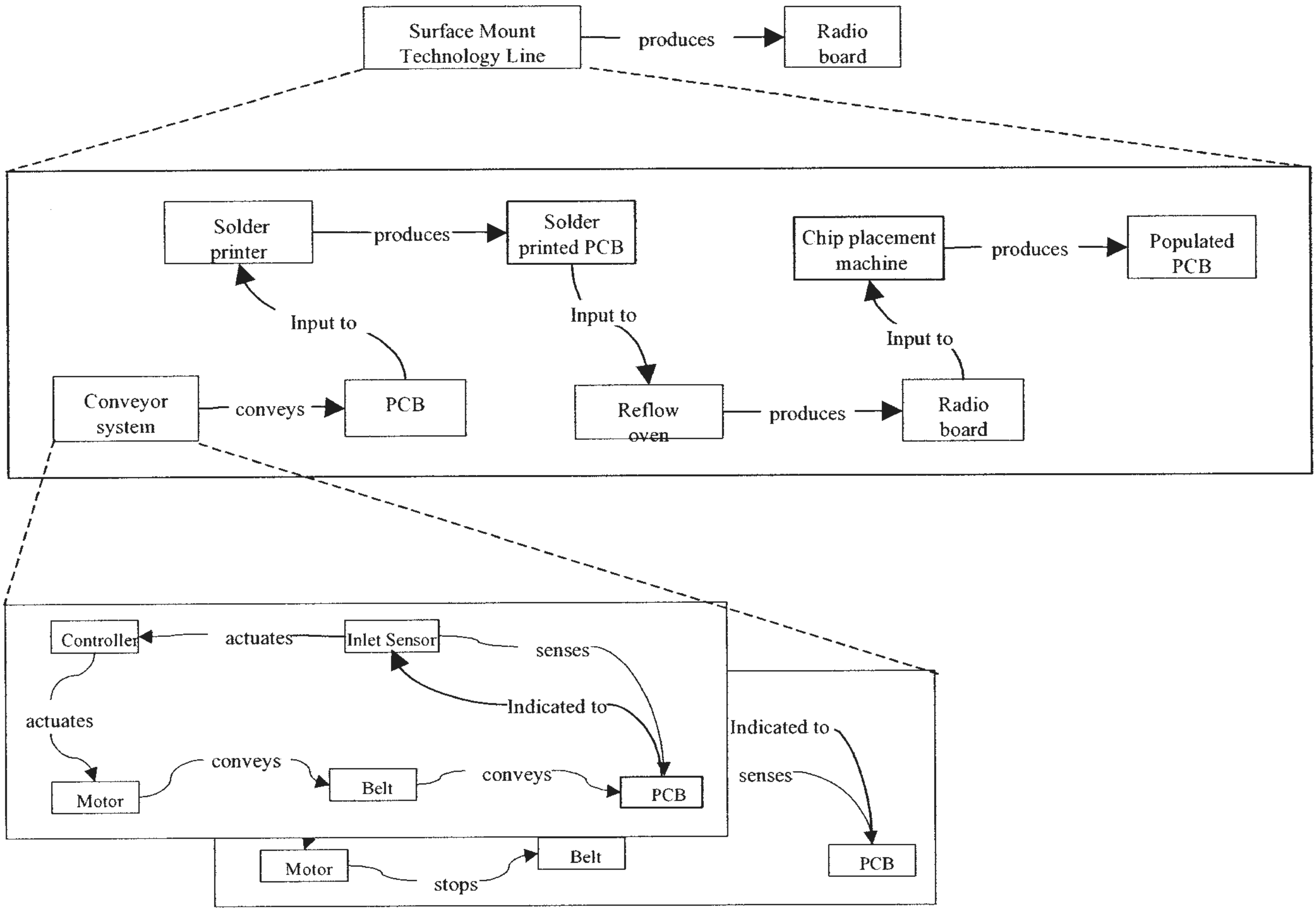

Fig. 10 FMAG model decomposition

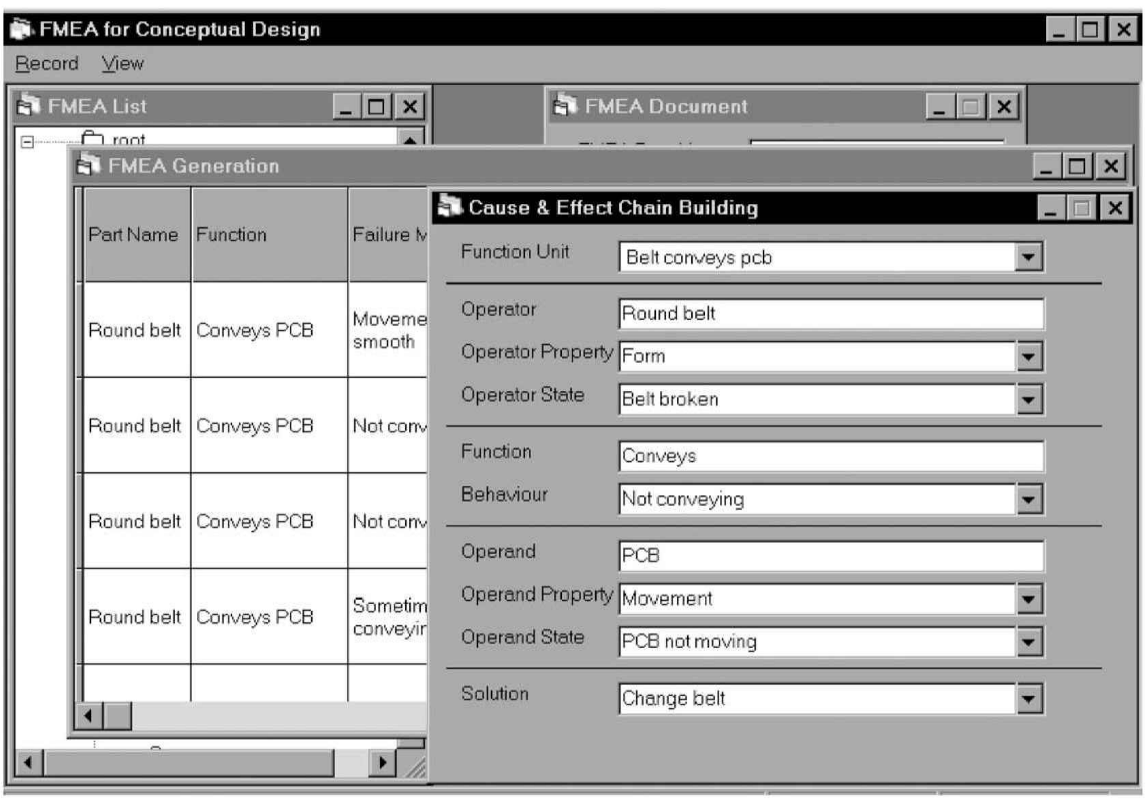

Fig. 11 Data entry in FMAG software

Not all generated items are valid. For example, for the case where a feeder positions a component for a pick-and-place process, the function unit is 'feeder–positions—component'. If the failure is 'feeder base not stable', the effect will be 'component location not consistent'. In another case where a nozzle places a component, the function unit is 'nozzle–positions–component'. If the failure is 'nozzle not moving into position', the effect will be 'component not placed'. Both examples above are valid results. However, the above data will cause the system to generate another result for the function unit 'feeder–positions–component'. Another effect for the failure 'feeder base not stable' is 'component not placed', which is not valid. Hence, different interpretations of the function 'positions' can cause some confusion in the results produced. This weakness is yet to be resolved.

However, looking at the results in total, the percentage of the invalid results is not high. Of the 339 FMEA items generated (each item is examined to determine whether the result is valid), 330 items were found to be valid (i.e. 97.3 per cent validity). Hence, the evaluation result strongly supported the validity of the two proposed basic assumptions used in the FMEA generation.

8 CONCLUSION

FMEA users face many difficulties due to the weakness of the current approach. The need to improve knowledge reuse at an early stage in design has prompted considerable research in FMEA. An effective way to improve the effectiveness of the FMEA is to automate the FMEA authoring process. This paper has reviewed modelling and reasoning techniques that are used to provide automatic FMEA generation.

The combination of IDEF3 and the functional diagram provide the basic model for the process. A 'knowledge fragment' reasoning approach is used to create cause-and-effect relationships. The reasoning is controlled by the precondition and postcondition relationships based on two basic assumptions, which lead to the formation of FMEA knowledge. The new approach has been tested and the case studies have shown promising results.

The content of the FMEA is naturally domain dependent, but it is believed that the methodology is generic and could be used for many applications. However, it is true that, as knowledge increases, the computational load will be more demanding and there is a need to study methods of addressing this aspect before contemplating large-scale practical implementations.

ACKNOWLEDGEMENTS

The authors would like to thank Mechanical and Manufacturing Engineering, Loughborough University and Motorola Technology Malaysia plc for supporting the research.

REFERENCES

1 **Chen, E. H. C.** *Failure Modes and Effects Analysis Training Manual*, 1996 (Chen Technology, Inc.).

2 BS 5760: Part 5: 1991 *Reliability of Systems, Equipment and Components. Guide to Failure Modes, Effects and Criticality Analysis* (*FMEA and FMECA*), 1991 (British Standards Institution, London).

3 **Dale, B. G.** and **Shaw, P.** Failure mode and effects analysis in the U.K. motor industry: a state-of-the-art study. *Qual. Reliability Int.*, 1996.

4 **Price, C. J.**, **Pugh, D. R.**, **Wilson, M. S.** and **Snooke, N.** Flame system: automating electrical failure mode and effects analysis (FMEA). In Proceedings of the Annual Reliability and Maintainability Symposium, 1995, pp. 90–95.

5 **Price, C. J.** AutoSteve: automated electrical design analysis. In Proceedings of the 1997 IEE Colloquium on Applications of Model-Based Reasoning, *Instn Electr. Engrs Colloq.* (*Dig.*), 1997, **338**, 4/1–4/3.

6 **Hughes, N.**, **Chou, E.**, **Price, C. J.** and **Lee, M.** Automating mechanical FMEA using functional models. In Proceedings of the 12th International Florida AI Research Society Conference, 1999, pp. 394–398 (AAAI Press).

7 **Atkinson, R. M.**, **Montakhab, M. R.**, **Pillay, K. D. A.**, **Woollons, D. J.**, **Hogan, P. A.**, **Burrows, C. R.** and **Edge, K. A.** Automated fault analysis for hydraulic systems. Part 1: fundamentals. *Proc. Instn Mech. Engrs*, *Part I: J. Systems and Control Engineering*, 1992, **206**(I4), 207–214.

8 **Hogan, P. A.**, **Burrows, C. R.**, **Edge, K. A.**, **Atkinson, R. M.**, **Montakhab, M. R.** and **Woollons, D. J.** Automated fault analysis for hydraulic systems. Part 2: application. *Proc. Instn Mech. Engrs*, *Part I: J. Systems and Control Engineering*, 1992, **206**(I4), 215–224.

9 **Bouti, A.**, **Ait Kadi, D.** and **Dhouib, K.** Automated manufacturing systems failure analysis based on a functional reasoning. *Information Technology for Modern Manufacturing*, In Proceedings of the 10th ISPE–IFAC International Conference on *CAD/CAM*, *Robotics and Factories of the Future CARs and FOF'94*, 1994, pp. 423–429 (OCRI Publications, Kanata, Ontario).

10 **Price, C. J.**, **Pegler, I. S.**, **Ratcliffe, M. B.** and **Sherwood, D.** *QPAC Handbook—How to Implement Integrated Quality Systems in a Factory Floor*, 1998 (Department of Computer Science, University of Wales, Aberystwyth).

11 **Eubanks, C. F.**, **Kmenta, S.** and **Ishii, K.** System behavior modelling as a basis for advance failure modes and effects analysis. In Proceedings of the ASME Design Engineering Technical Conference and Computers in Engineering Conference, Irvine, California, 1996 (American Society of Mechanical Engineers, New York).

12 **Eubanks, C. F.**, **Kmenta, S.** and **Ishii, K.** Advanced failure modes and effects analysis using behavior modelling. In Proceedings of the ASME Design Engineering Technical Conference and Design Theory and Methodology Conference, Sacramento, California, 1997 (American Society of Mechanical Engineers, New York).

13 **Kusiak, A.** *Engineering Design—Products, Processes and Systems*, 1999 (Academic Press, San Diego, California).

14 **Hunt, J. E., Pugh, D. R.** and **Price, C. J.** Failure mode effects analysis: a practical application of functional modelling. *Appl. Artif. Intell.*, 1995, **9**(1), 33–44.

15 **Gero, J. S., Tham, K. W.** and **Lee, H. S.** Behaviour—a link between functional and structure in design. In Proceedings of the IFIP WG 5.2 Working Conference on *Intelligent Computer Aided Design*, 1991 (Elsevier, Amsterdam, The Netherlands).

16 **Hubka, V.** and **Eder, W. E.** *Theory of Technical Systems: A Total Concept Theory for Engineering Design*, 1988 (Springer-Verlag, Berlin).

17 **Pahl, G.** and **Beitz, W.** *Engineering Design, A Systematic Approach*, 2nd edition, 1996 (Springer-Verlag, London).

18 **Ulrich, K. T.** and **Eppinger, S. D.** *Product Design and Development*, 1995 (McGraw-Hill, New York).

19 National Institute of Standards and Technology (1993). *Integrated Definition for Function Modelling (IDEF0)*, 1993 (Federal Information Standards Publications); http://www.idef.com/Downloads/pdf/idef0.pdf.

20 **Dorador, J. M.** Product and process information interaction in assembly decision support system. Doctoral Thesis, Loughborough University, 2001.

21 **Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T.** and **Perakath, B.** Information integration for concurrent engineering (IICE) IDEF3 process description capture method. Report, Knowledge Based System Inc., 1995.

22 **Chandrasekaran, B.** and **Josephson, J. R.** Representing function as effects: assigning functions to objects in context and out. In Proceedings of the AAAI Workshop on *Modeling and Reasoning with Function*, 1996, pp. 30–37.

23 **Maher, M. L.** and **Pu, P.** *Issues and Application of Case-Based Reasoning in Design*, 1997 (Lawrence Erlbaum Associates).

24 **Kato, Y., Shirakawa, T.** and **Hori, K.** Department of Advanced Interdisciplinary Studies, University of Tokyo, 2002; http://www.ai.rcast.u-tokyo.ac.jp/~yoshi/papers/kes2002.pdf.

