(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2012 (01.11.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/146908 A1

(51) International Patent Classification:
G06F 17/50 (2006.01) *G05B 23/02* (2006.01)

(21) International Application Number:
PCT/GB2012/050799

(22) International Filing Date:
12 April 2012 (12.04.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1107160.2 28 April 2011 (28.04.2011) GB
11275068.2 28 April 2011 (28.04.2011) EP

(71) Applicant *(for all designated States except US)*: **BAE SYSTEMS PLC** [GB/GB]; 6 Carlton Gardens, London SW1Y 5AD (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **SNOOKE, Neal, Andrew** [GB/GB]; University of Aberystwyth, Llandinam Building, Aberystwyth, Ceredigion SY23 2AX (GB).

(74) Agent: **BAE SYSTEMS PLC, Group IP Dept**; PO Box 87, Farnborough Aerospace Centre, Farnborough, Hampshire GU14 6YU (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: AUTOMATED METHOD FOR GENERATING SYMPTOMS DATA FOR DIAGNOSTIC SYSTEMS

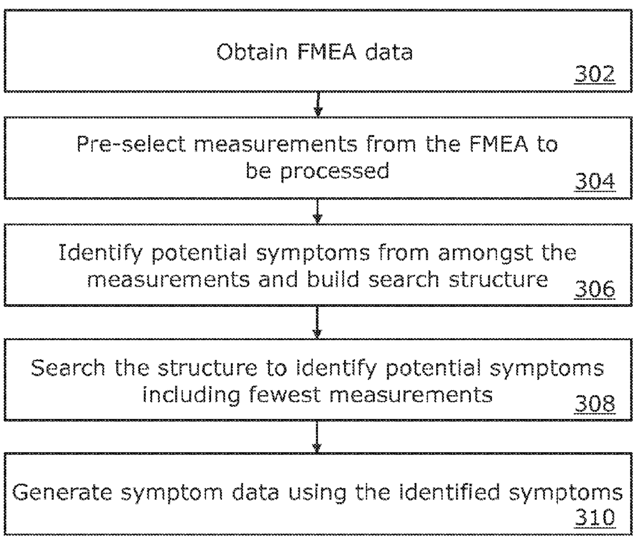

Fig. 3

(57) Abstract: A method of generating symptom electronic data for use in a diagnostic system (210). The method includes: i) obtaining (302) Failure Modes and Effects Analysis (FMEA) data for a target system, the data including a plurality of observations, each said observation including a set of at least one measurement associated with a fault state or a nominal state of the target system; ii) selecting (304, 306) a first said observation associated with a nominal state; iii) selecting (304, 306) a second said observation associated with a fault state; iv) identifying (306) at least one measurement in the second observation that is/are not present in the first observation, and v) generating the symptom data based on the identified at least one measurement.

WO 2012/146908 A1

AUTOMATED METHOD FOR GENERATING SYMPTOMS DATA FOR DIAGNOSTIC SYSTEMS

The present invention relates to diagnostic systems and in particular to generating symptom data for diagnostic systems.

The diagnosis of complex systems can be a challenging task and model based diagnosis has been widely applied to produce both on-board and off-board (workshop based) systems that assist in the detection and isolation of faults. The majority of these known systems use executable models of the systems (either with or without fault models), simulated in parallel with the working system. The model is supplied with the actual inputs to the system and abductive reasoning is performed to detect faults based on discrepancies between the predicted and observed system behaviour. Such known systems are sophisticated and can potentially detect and isolate a wide range of faults, including in some cases novel faults and arbitrary multiple faults. However, there are some drawbacks. For instance, execution complexity and model accuracy can be technical challenges and validation of the performance can be an issue where strict regulation approval is required.

The paper “Generating a diagnostic system from an automated FMEA” (in Proceedings of the Annual Conference of the Prognostics and Health Management Society, 2009, San Diego, CA, USA, September 27 – October 1 2009, pages 1 – 12) authored by the present inventor outlines a software-based graphical tool for assisting an engineer with investigating the diagnostic ability of a system or product based on existing or additional sensing. Although the paper briefly discusses symptom generation for use in FMEA, it does not propose any method of automatically generating data for use in a diagnostic system. The paper therefore does not propose any solution to the technical problem of automated symptom generation, which can then be used to solve technical problems relating to system diagnosis and unassisted fault detection and isolation.

WO2006/002527 discloses a symptom generation technique, but one that does not automatically generate symptoms that inherently include all the necessary operating state information required to detect faults and so would require operator intervention to achieve this. The simulation must be configured to match the state of the system and requires an operator to recognise anomalous behaviour and then configure the simulation to achieve the relevant operating state before candidate faults can be deduced.

JP2003228485 describes a diagnosis rule generation method involving a diagnosis/fault tree, but does not take into account operating modes or states of the system (i.e. it does not identify or use relationships between observations associated with nominal and fault states). The method transforms the tree into an FMEA and then requires engineer effort to generate symptoms based on the FMEA.

Embodiments of the present invention are intended to address at least some of the problems discussed above, in particular the technical problem of automated symptom generation, which can then be further used to solve technical problems relating to system diagnosis and unassisted fault detection and isolation. Embodiments of the approach described herein are focused on producing a pre-compiled diagnostic system based on a set of easily evaluated symptoms that do not require complex simulation when the system is in use. This solution was inspired by the need to capture failure mode and effect information from manually generated FMEA forms for entry into an existing proprietary Bayesian network based diagnostic system. An automatically generated FMEA can be used to reduce effort and increase the speed of FMEA production. The automated version of an FMEA is usually more detailed in terms of the effects and behaviour and therefore manual symptom generation can be more time consuming; however, the symptoms can be more comprehensive. The approach described herein can automate the generation of symptoms from the automated FMEA output.

According to a first aspect of the present invention there is provided a (computer-implemented) method of generating (electronic) symptom data for use in a diagnostic system, the method including:

i) obtaining Failure Modes and Effects Analysis (FMEA) (electronic) data for a target system, the data including a plurality of observations, each said observation including a set of at least one measurement associated with a fault state or a nominal state of the target system;

ii) selecting a first said observation associated with a nominal state;

iii) selecting a second said observation associated with a fault state;

iv) identifying at least one measurement in the second observation that is/are not present in the first observation, and

v) generating the symptom (electronic) data based on the identified at least one measurement.

The method may include (prior to step ii)):

i') identifying said observations from the FMEA data associated with a fault state that indicates failure of a function of the target system, and

using these identified observations for the selection of the first observation and the second observation.

The method may include identifying said observations from the FMEA data associated with all fault states that indicate failure of each said function of the target system.

For each said second observation OBS^M_x related to a particular failure mode M, defining an ordered set $\overline{N}$ comprising sets of said measurements of the second observation that are abnormal with respect to the measurements of the first observation. The method may further include (following step iv)): iv') assigning an index value to the measurements in the ordered set $\overline{N}$. The index value (for a said measurement "o" being processed during an iteration "i" of the method) may be defined by:

$$\mathrm{Index}(o) = \sum_{i=1..|\overline{N}|} \begin{cases} 2^{i-1}, & \text{if } o \in \overline{N}_i \\ 0, & \text{otherwise} \end{cases}$$

The steps ii), iii), iv) and iv') may be repeated for several combinations of said observations from the Failure Modes and Effects Analysis data, resulting in a set of potential symptoms E_p, each comprising at least one measurement from every said set in $\overline{\mathrm{N}}$, with each member of the potential symptom set E_p being associated with a said index value. A total index value of each said member may comprise a valid symptom being equal to 2^x, where x is a number of elements in the set $\overline{\mathrm{N}}$.

The method may further include creating a search structure including the potential symptom set. The search structure may comprise a binary search tree structure, with a position of a said measurement of the potential symptom set in the search tree structure being based on a decomposition of the index value by a power of 2. The method may further include performing a search on the search tree structure to identify members of the potential symptom sets having a fewest number of said measurements. The search may comprise a depth-first search. The identified members of the search tree structure may be used to generate the symptom data.

The step v) of generating the symptom data may include generating data relating to at least one said symptom that, during a diagnostic operation, is detected by the at least one measurement.

According to another aspect of the present invention there is provided a method of diagnosing a target system including:

inputting symptom data generated by a method substantially as described herein into a diagnostic system, and

using the symptom data in the diagnostic system to identify faults in the target system.

According to other aspects of the present invention there are provided computer program products comprising computer readable medium, having thereon computer program code means, when the program code is loaded, to make the computer execute methods substantially as described herein.

According to a further aspect of the present invention there is provided apparatus configured to execute a method substantially as described herein.

According to yet another aspect of the present invention there is provided a method of assisting with system diagnosis or FMEA substantially as described herein.

According to yet another aspect of the present invention there is provided a method of adding measurement devices to a system, the method including using the symptom data generated substantially as described herein to determine a type or location of a said measurement device in the system.

The techniques described herein can use the results of an automated Failure Modes and Effects Analysis (FMEA) to generate qualitative symptoms for use in an automated diagnostic system. The symptom generation does not usually require any additional modelling effort from the engineer beyond the models required to produce the FMEA. The symptoms may then be used as the basis of an automatically generated on board diagnostic system, or as tool to assist in qualitative “diagnosability” studies such as sensor placement choices during system design.

The automated FMEA can be produced from a system schematic and component model library using qualitative model based simulation techniques, and has proven to be successful in a number of engineering application areas from the automotive and aerospace domains. The comprehensive analysis undertaken during FMEA can provide a great deal of diagnostic information concerning the effects and observations that are related to possible system faults. However, FMEA can be focussed on providing a worst case comparison between known nominal and abnormal functional effects with respect to each failure mode, whereas a symptom based diagnostic system needs to rely only on actual observations without external knowledge of expected behaviours. The present technique can process the detailed information produced during the FMEA into a set of symptoms based on qualitative observations.

Whilst the invention has been described above, it extends to any inventive combination of features set out above or in the following description. Although illustrative embodiments of the invention are described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to these precise embodiments. As such, many modifications and variations will be apparent to practitioners skilled in the art. Furthermore, it is contemplated that a particular feature described either individually or as part of an embodiment can be combined with other individually described features, or parts of other embodiments, even if the other features and embodiments make no mention of the particular feature. Thus, the invention extends to such specific combinations not already described.

The invention may be performed in various ways, and, by way of example only, embodiments thereof will now be described, reference being made to the accompanying drawings in which:

Figure 1 is a schematic drawing of a model-based on board symptom generation system;

Figure 2 is a schematic drawing of a computing device configured to provide an embodiment of the symptom generation system;

Figure 3 is a flowchart showing a symptom generation process performed by the system;

Figure 4 illustrates schematically part of the symptom generation process;

Figure 5 illustrates schematically FMEA system states, and

Figure 6 illustrates schematically another part of an example operation of the process.

Figure 1 illustrates the main steps in the process of producing detectable symptoms for the on-board diagnostic system. Model-based generation of FMEA 102 has been well-documented and is in daily use in industry to produce FMEA reports on automotive electrical systems. It uses information regarding the structure 104 and functions 105 of the system, simulates the qualitative system behaviour 106 for all potential component failures (via a model based simulation 108 of the system) and uses functional reasoning to abstract the high level consequences for each possible failure. The consequences of every failure are reported to the designers, via the produced FMEA results 110, and they can decide the steps needed to improve the safety and reliability of the system.

The symptom generation process 112, based on the FMEA results 110, will be described in detail below. The FMEA results provide a link between the system failures and the component faults that caused them and can be used to produce a list 114 of system symptoms/problems. For service bay diagnostics, these links can be used to end the cause of a problem with the system. However, in on-board diagnosis, the known information is limited to sensor values and system state, and it is necessary to decide how and when to use observable behaviour both to decide that there is a problem, and to deduce the root cause of that problem. Embodiments of the present invention use model-based reasoning to generate a set of characteristic symptoms to monitor that will provide the maximum information about the state of the system.

The symptom/problem list can be received by an on-board diagnostic system 116 (which can, in some cases, also receive symptom information 117 based on an engineer's input). One of the limitations of model-based reasoning is that it can only deal with problems that are within the scope of the modelling. For example, if problems are caused by cross-component interactions that are not modelled, then they would not be included in the set of symptoms to monitor. The generated set of symptoms to be monitored can be augmented by further symptoms that compensate for the limitations of the chosen modelling. The symptoms can be included in a Bayesian network based diagnostic system, as described below. Although the symptoms can be used in a Bayesian system in this way, if there are probabilistic elements associated with uncertainty of measurements or reliability of sensors, it will be appreciated that alternative methods are possible. For example, the generated symptoms could be used directly (as generated below) to provide a ranked list of likely faults in a target system.

The symptom/problem is list is also received by a coverage analysis process 118, which can produce data 120 representing symptom coverage. Ideally, the on-board diagnostic system 116 would be able to detect and isolate every possible component fault that could occur. In practice, on-board systems have a limited set of sensors, and those sensors are often decided before an on-board diagnostic system is planned. It is, however, easy to modify the visibility of any model parameter or simulation variable to the symptom generation. The present inventors have developed a tool that exploits this to allow investigation and analysis of the relationship between measurement availability and diagnosability, presenting the information to the engineer as a visual cluster matrix together with summary information. These techniques and tools can be used during early design to quickly assess the impact of different sensor selection strategies.

Figure 2 schematically shows a computing device 200 including a processor 202 and a memory 204. Other common elements of the computer, e.g. display, external storage, communications means and user interface means, are well known and are not shown. The memory includes FMEA data 206 (which can be generated by existing FMEA methods/tools) and an application 208 for producing symptom data according to an embodiment of the present invention. The symptom data can then be used by a diagnostic application 210 for diagnosing faults in a target system, for example. It will be appreciated that in many cases the symptom generation application 208 and the diagnostic application 210 may be executed on different computing devices, or may be in communication over a network, etc. Alternatively, the applications 208 and 210 may be integrated.

Below is a description of how the generated symptom fault sets can be used to create a simple diagnostic system, followed by the results of applying the technique to realistic sized systems. There are several advantages of the system described here compared to one which performs on-board simulation to do diagnosis:

- The diagnostics can be checked and verified prior to deployment, which is necessary to meet stringent certification requirements.

- The simulation effort need be carried out once only, and is part of existing design analysis requirements, and does not need to be done on the on-board hardware. This means that the diagnostic system requires relatively simple on board processing. Additional symptoms can be inserted manually based on experience if necessary. This allows symptoms based on specialist sensors outside of the modelling domain of the system to be included such as a temperature sensor measuring electrical component temperatures. In addition, outputs of components or subsystems that include their own black box diagnostic capabilities that cannot be modelled can be included in the diagnostic system. Further symptoms can be included (or possibly excluded) where for pragmatic reasons modelling simplifications exist in the models used for the design analysis.
- The Bayesian diagnostic system allows adjustment of the weighting of symptoms to reflect any uncertainty regarding the applicability of a symptom, for example where measurements might be considered less reliable.
- The diagnostic system's capabilities can be characterized in advance, and the sensing requirements and cost weighed against the diagnostic and fault isolation capability of the system. An engineer can experiment with sensor selection and assess the capability of the resulting diagnostic system.

A formal description of the process of generating a model-based FMEA report abstracted by function shall now be given. This will then be used as a basis for the novel method of symptom generation.

<u>FMEA generation</u>

An FMEA report is produced by comparison of system function and behaviour for nominal and failure mode operation. The system is exercised through a number of states using a scenario defined by an engineer that (in general) activates each of the system functions and major functional operating modes. The following paragraphs develop a notation for the FMEA results to be used for an example embodiment of the symptom generation process.

OBS is defined as a finite set of first order sentences representing all measurements (outputs or inputs, or observable component states) available from a target system. Some members of OBS may be complex expressions, for example to create “virtual sensors”; however, the term “measurement” is used to refer to a single sentence because the majority contain a simple proposition that compares the value of a measurable quantity. Typical sentences in the qualitative representation of the system used in this work might be temperature = high or switch.position = on although it could also be a component state, software variable or any other observable. A scenario, SCN, will provide an ordered sequence of n input states for the system where SCNn $\subset$ OBS. That is, SCNn defines a set of triggers. A simulation is performed for each SCNn, and for systems that reach a steady state, an observation is produced (containing all available measurements). The notation OBSn is used to indicate the subset of OBS produced (by simulation) for SCNn with no faults present.

A target system is comprised of a set of constants COMPS representing the components of the system. Each component has a set of mutually exclusive failure modes Modes(c) where c $\in$ COMPS. Single faults are generally used for FMEA generation, not only because of the effort involved in considering the results of multiple faults, but also because most design issues are highlighted from the single fault cases. The complete set of component failure modes M for a system is thus:

$$\mathcal{M} = \bigcup_{\forall c \in COMPS} \text{Modes}(c) \qquad (1)$$

The simulation can support any number of simultaneous failures and on occasion an engineer may decide to include specific categories of multiple failure. Throughout this specification, M is used to represent a (simultaneous set of) component failures considered as a failure mode in an FMEA. M = Ø is considered as the no failure case, and usually M contains a single component failure mode although generally $M \subseteq \mathrm{M}$ ensuring only one failure mode from each component:

$$(\forall m, \forall n \in M, n \neq m, \forall c \in COMPS)$$

$$(m \notin \mathrm{Modes(c)} \vee n \notin \mathrm{Modes}(c)) \qquad (2)$$

The notation OBS^M is used to refer to the set of observations obtained by simulation of the system for SCN with faults M present and OBS_n^M thus provides concise notation for a failure mode observation, a single set of measurements for SCNn.

Function Model

A functional interpretation model can be used to allow the results of a component based qualitative simulation to be abstracted into effects meaningful to an engineer for the purpose of FMEA generation. The functional model also has a role in the ability to generate diagnostics from the simulation results produced during the FMEA because it can determine the relevance of measurements by operating state, thus allowing symptom generation using only a selection of representative system states.

An engineer identifies the system functions, $\mathcal{F}$, using a known Functional Interpretation Language (FIL) (see, for example, J. Bell, N. A. Snooke and C. J. Price, A language for functional interpretation of model based simulation. *Advanced Engineering Informatics,* 21(4): 398- 409, Oct 2007). The functional description can be as simple or sophisticated as required, although generally functions are easily identified – even if the functions themselves are internally complex – by defining the triggers required to activate the function and the effects required for the function to be considered as achieved. The triggers and effects are linked to fragments of system behaviour for any system that implements the function. For the purpose of this specification it is only necessary to understand that each system function, $fn \in \mathcal{F}$, can be in one of four possible states provided by the predicates Ac(fn), In(fn), Fa(fn), Un(fn), that represent the function having been Achieved, Inoperative, Failed or Unexpected. These function states are defined in terms of trigger (Tr) and effect (Ef) propositions, grounded in a subset of the possible system measurements within the FIL as follows:

$$\mathrm{Ac}(fn) \equiv \mathrm{Tr}(fn) \wedge \mathrm{Ef}(fn) \qquad (3)$$

$$\mathrm{In}(fn) \equiv \neg\mathrm{Tr}(fn) \wedge \neg\mathrm{Ef}(fn) \qquad (4)$$

$$\mathrm{Fa}(fn) \equiv \mathrm{Tr}(fn) \wedge \neg\mathrm{Ef}(fn) \qquad (5)$$

$$\mathrm{Un}(fn) \equiv \neg\mathrm{Tr}(fn) \wedge \mathrm{Ef}(fn) \qquad (6)$$

Some intuition as to the type of knowledge captured can be seen in the illustration below of a “high_beam” function associated with an automotive lighting application:

```
FUNCTION high_beam  {
        ACHIEVES light_road_ahead
        BY switch_on_main_beam TRIGGERS main_lamps_lit }

TRIGGER Switch_Lighting_S100.Switch2Pos  ==  'pos1'
                AND Switch_LightingS101.Switch3Pos  =  'main_lights'
        IMPLEMENTS switch_on_main_beam

PURPOSE light_road_ahead  {
        DEPLOYMENT vehicle
        DESCRIPTION 'lights the road ahead'
        FAILURE_CONSEQUENCE 'impared night vision'
                SEVERITY 8 DETECTABILITY 6 }

EFFECT Headlamp_RH_A100.main.i  ==  'MEDIUM'
                AND Headlamp_LH_A101.main.i  ==  'MEDIUM'
        IMPLEMENTS main_lamps_lit
        UNEXPECTED_CONSEQUENCE 'may dazzle oncoming traffic at night'
        SEVERITY 9 DETECTABILITY 6
```

Any undefined identifiers are references to values provided by an associated system simulation. For example, the proposition Switch_Lighting_S100. Switch2wayPosition == ‘pos1’ is associated with a switch input position and provides the trigger for the function and the observations Headlamp_RH-A100.main.i == 'MEDIUM' and headlamp_LH_A101.main.i == 'MEDIUM' evaluate current flow in electrical components of the system and are associated with the required effect (the FIL uses == as the equivalence operator and = for assignment). Purpose information is not used in symptom generation but captures the role of the function within a deployment environment. Generally, the functional model will be a relatively simple identification of each system function, how it is activated and what effects are to be expected.

FMEA Abstraction

The engineer-level FMEA report is generated by:

- simulating the system with a variety of failure modes $M_1;M_{2,..}$
- evaluating the state of system functions for each element in $OBS^M{}_i$
- reporting failed and unexpected functions using the definitions in equations (5) and (6).

The FMEA report for simulation step SCN_n of fault M will contain information relating to the following set of abnormal functional results:

$$RfnAb_{M\,n} = \{fn \mid Fa(fn) \text{ evaluated for } OBS^M{}_n \vee Un(fn) \text{ evaluated for } OBS^M{}_n\} \quad (7)$$

In addition, the related sets of failed and achieved functions are of interest for symptom generation:

$$RfnFa^M{}_n = \{fn \mid Fa(fn) \text{ evaluated for } OBS^M\} \quad (8)$$

$$RfnAc^M{}_n = \{fn \mid Ac(fn) \text{ evaluated for } OBS^M{}_n\} \quad (9)$$

The functional information provides adequate information to generate a traditional FMEA; however, specific issues can also benefit from additional explanation with detailed behavioural abnormalities identified by comparison of the nominal and failure observations:

$$RobsAb^M{}_n = \{o \in OBS^M{}_n \mid \in OBS_n\} \quad (10)$$

An example entry generated in an FMEA output form is shown in the table below (a fragment from a typical FMEA), showing the behaviour and functional effect of a wire fracture in the electrical system of a car.

The following table lists the results from the automatically generated FMEA

Item	Behaviour		Failure mode			Failure Causes	Sev	Det	Occ	RPN
F1	At Off, High beam (Step_3)					W28 - fracture	8	6		48
	Observable	Value	Function	Effects	S,D					
	'W28 Flow'	inactive (active expected)	'high_beam' failed	impared night vision	8, 6					
	'W1 Flow'	inactive (active expected)								
	'S51 Flow'	inactive (active expected)								
	'Instrument_Pack_J100 main_beam_indicator'	off (on expected)								

Once these results are considered acceptable by an engineer symptom generation can be carried out. A prerequisite for FMEA production is that there should be no abnormal functions in the absence of a failure:

$$(\forall n)(R\mathrm{fn}Ab_n \Leftrightarrow \varnothing) \qquad (11)$$

If proposition (11) is not satisfied for some OBS then the system has failed to implement its function description and this design verification issue should be addressed before any meaningful FMEA or symptom generation can be produced.

<u>Symptom generation</u>

When vehicle on-board diagnostics are decided by engineers, they typically identify a sensor or set of sensors to observe, and an operating state of the system in which the observations can be made. For example, if there is no flow of fuel at sensor FS when pump P is ON, then pump P is not working. The simulations carried out to produce an automated FMEA provide a representative subset of the possible failure and nominal system states. These simulations can be used to identify symptoms characteristic of specific faults, but to produce symptoms that diagnose faults over the majority of system states requires generalisation of the effects of faults.

The automated symptom generation technique described herein (and embodied in application 208) does not require explicit external knowledge concerning the relevance of measurements to either faults or functions to compensate for the missing state information inherent in the FMEA; rather, this information can be extrapolated from analysis of behavioural consistency covering all of the faults and all of the system states encountered during the entire FMEA. Given a reasonably comprehensive FMEA that exercises all system functions and includes faults for the majority of components, the aim of the system is to produce symptoms that are general enough to cover a great deal of the system state that is not explicitly included in the FMEA without producing spurious symptoms. The functional model (also used to interpret the FMEA) is an enabling concept that indirectly provides abstract guiding knowledge facilitating extrapolation of symptoms to system states that were not present in the FMEA, as will be described below.

A symptom is defined as a tuple S = (E, F) such that E is a first order sentence referred to as a “symptom expression” that when satisfied indicates F = $\{M_1, M_2;\ldots\}$ as “symptom faults”. A set of satisfied symptoms forms the diagnosis candidates $D \subset M$. E(s) evaluates E on a specific system state $s \in$ OBS. The term “simpler symptom” is used to refer to a symptom that has fewer measurements required to evaluate E. A more complex symptom requires a greater number of measurements than a simpler one. In this specification, E is always a simple conjunction of equivalence propositions that represent measurements, and we use E to represent these individual measurements and therefore the simplicity measure for any symptom is |E|. For example if E = (switch1 <-> closed $\wedge$ wire1.i <-> positive) then E = {switch1 <-> closed; wire1.i <-> positive)}. If |F| is smaller for symptom S_1 than S_2 then we refer to S_1 as more specific than S_2 and S_2 as more general then S_1.

The inventors found empirically that by forcing the generation of complex symptoms that they can represent artefacts in the modelling, or artefacts due to the incomplete state availability of real FMEA analyses. Therefore, a general principle used is to generate the simplest symptoms that provide the required fault detection/isolation. An aim of the symptom generation is to produce a symptom set S = $\{S_1; S_2; \ldots\}$ that will provide a diagnosis as a set of possible faults D, such that if M is a given failure mode, then $M \in D$ while avoiding spurious diagnoses $M \notin D \wedge D \neq \emptyset$. No diagnosis of a fault D = Ø is preferable to a spurious diagnosis but given a fault M, the more operating time or states that $D \neq \emptyset$ the more powerful the fault detection, and the smaller $|D|$ the better the diagnostic system fault isolation. These constraints define the basic goals of embodiments of the symptom generation technique.

At first glance, the FMEA appears to directly provide symptoms simply by associating abnormal behaviour $RobsAb_n$ and failure modes M. While this may provide a usable symptom for some measurements, the majority of $RobsAb_n$ measurements remain within the range of values encountered during nominal operation. Therefore, most symptoms actually require one or more nominal measurements in addition to a subset of $RobsAb_n$ to provide a symptom that does not spuriously indicate faults. A fault may cause many abnormal observations and since it is beneficial for $|E|$ to be as small possible for a given level of diagnosability and fault isolation, the abnormal observations in $|E|$ may need to be a subset of those available in a specific abnormal observation. When manually creating symptoms from a hand generated FMEA, an engineer would make these selections and create conditional symptoms (discussed below) based on extensive system knowledge. The technique described herein automates this process and produces a consistent and comprehensive set of symptoms.

Figure 3 is a flowchart showing steps that can be performed by an example symptom generation application 208. It will be understood that in some cases that some of the steps may be re-ordered or omitted. At step 302 FMEA data 206 is obtained, typically by loading a file or receiving data over a network. An example of the format of FMEA data is given the table above, but it will be understood that variations to the content are possible.

At step 304 the symptom generation application 208 may pre-select measurements to be processed in order to generate the symptom data, which can improve efficiency of the process. The selection may be done based on the techniques described in the “Avoiding over generalisation” and “Associating measurements to function” sections below. Generally, an aim is to select only measurements that are causally related to faults; ones that only indicate a nominal state will not be so useful for diagnosis.

At step 306 the process identifies potential symptoms from amongst the measurements being processed and builds a data structure (such as a tree) including these potential symptoms. At step 308 the data structure constructed at step 306 is traversed/searched to identify potential symptoms that include the fewest number of measurements. This can be done for the reasons set out, and using the techniques described in, the "Identification of symptomatic measurements" and subsequent sections below. Generally, an aim is to identify the potential symptoms including the shortest list of measurements as these may be usable to diagnose more than one fault in an efficient manner.

At step 310 the symptoms identified in step 308 are used to generate the symptom data. It will be appreciated that the exact format and contents of this symptom data can vary, e.g. for compatibility with a particular diagnostic system. The data can then be stored or transferred for use by a diagnostic system, such application 210, or for other purposes.

Below is a more formal description of a strategy for generating symptoms assuming all operational system states are represented in the automated FMEA, prior to dealing with the more realistic situation where only a selection of key functional states are represented.

Identification of symptomatic measurements

Every component fault M considered by an FMEA results in several sets of measurements OBS^{M}_{n} as a response to the steps in the scenario. To produce an FMEA output form, these failure mode observations are abstracted using the functional model to create a description of the functional discrepancies potentially caused by the fault. The detailed failure mode observations are used directly to generate symptoms. One or more measurements may be conjoined to form a symptom; however, they must satisfy two constraints. First, a symptom should only be present when the associated fault is present. Second, it is desirable to detect as many faults as possible in as many operating states as possible. This implies symptoms should be general and therefore contain the fewest measurements, such that the first constraint is satisfied. Exhaustive searching for measurements that satisfy the constraints becomes intractable for systems with large numbers of measurements, but this section describes a method that rapidly locates all of the simplest solutions for very large numbers of measurements.

Given a failure mode observation, $s \in OBS^M$, the aim is to find sets of measurements, $E \subseteq s$, where $|E|$ is a small as possible, such that:

$$E = \{R \subseteq s \mid \nexists n : R \subset OBS_n\} \qquad (12)$$

For each fault observation $OBS^M{}_x$ related to a failure mode M, a set $\overline{N}$ is defined that itself contains sets of measurements that are abnormal with respect to each of the nominal failure observations. Each failure mode is considered individually so the fault mode superscript is omitted from the notation of N in the following description:

$$\overline{N}_x = \{ OBS^M{}_x \setminus OBS_1; OBS^M{}_x \setminus OBS_2; \ldots OBS^M{}_x \setminus OBS_{|SCN|}\} \qquad (13)$$

where $|SCN|$ is the number of distinct steps in the scenario.

Any symptom must contain at least one measurement from every element of $\overline{N}$ to satisfy the requirement that the symptom does not contain only measurements that exist during nominal operation. Definition (12) can thus be written:

$$E_x = \{R \subseteq OBS^M{}_x \mid \forall \overline{N}_i \in \overline{N} : \exists o \in N_i \cup R\} \qquad (14)$$

To efficiently select measurements, an index is assigned to each member of OBS^M according to which members of N it appears in. N is considered as an ordered set and each member is assigned an index value from a power 2 sequence. This facilitates a tree structured search that locates simple symptoms rapidly and allows the majority of longer symptoms to be ignored. Each measurement o is assigned its index in the following way:

$$\text{Index}(o) = \sum_{i=1..|\overline{N}|} \begin{cases} 2^{i-1}, & \text{if } o \in \overline{N}_i \\ 0, & \text{otherwise} \end{cases} \qquad (15)$$

The index values assigned to each measurement define a binary tree structure for the measurements, with the position of the measurement in the tree based on a decomposition its index by powers of two. For example, if L and R specify either the left or right branches of the tree where the left branch is chosen if the index contains a power of two element, then for a tree of depth 3, a measurement with index 7 is on branch LLL (4+2+1) and a measurement with index 5 (4+1) is on branch LRL and a measurement index 1 on branch RRL (1).

Figure 4 illustrates the assignment of indices for a small example with 3 scenario states and 6 possible measurements a-f. The nominal observations are shown at the top of each column representing a scenario state and the possible failure observations for the failure are shown below.

$\overline{N}_i$ for a specific failure observation is generated by removing the nominal mode observations for scenario step i from the given failure observations.

The ordering of the measurements is depicted by a matrix using rows to represent measurements and columns to represent each element of $\overline{N}$. The sum of the abnormal observation set indices for each row provides the order to search the measurements.

Since $|\overline{N}| = |SCN|$ and is related to the number of system functions, the index values have been small enough to compute explicitly for the systems we have applied the method to. A sorting algorithm that performs a sequence of partial sorts of s based on membership of each element in $\overline{N}$ would be equivalent for larger scenarios, however it is doubtful that an FMEA with such an excessive number of scenario steps would be useful, and a better solution might be to divide the analysis into subsystems. The case where index(m) = index(n) for two measurements is considered below.

The symptoms are now generated by traversing the ordered list of measurements, including new measurements until a symptom is formed that includes at least one measurement from each set of $\overline{N}$:

1. Initially set $k = |\overline{N}|$
2. Each measurement m_p contained in $\overline{N}_k$ is considered as part of a new potential symptom expression, ε_p.

Consider each ε_p in turn:

3. *Completed symptoms check.* If ε_p contains at least one element from every set in $\overline{\mathrm{N}}$ (i.e. ε_p where $\sum_{\forall m \in \varepsilon p}$ index(m) = $2^{|N\neg|-1}$) then ε_p is a completed symptom. ε_p is converted into a symptom expression by conjoining all the elements E = $\Lambda_{\forall m \in \varepsilon p}\ m$. Consider any other ε_p possibilities.

4. An $\overline{\mathrm{N}}$ index, *j,* is chosen for ε_p such that $1 \leq j < k$ and $(\forall m \in \varepsilon_p)(m \notin \overline{\mathrm{N}}_j)$. In addition, if $j < i < k$, then $(\nexists\ i)(\forall m \in \varepsilon_p)(m \notin \overline{\mathrm{N}}_i)$. That is, $\overline{\mathrm{N}}_j$ must be the element in $\overline{\mathrm{N}}$ not already represented by the symptom being constructed.

5. *Dead End Symptom check.* If no *j* can be found that satisfies the constraints in step 4 then ε_p cannot form a symptom and is removed from consideration. Consider any other ε_p possibilities.

6. ε_p is used as the basis of a new set of partial symptoms, $\varepsilon_{p'}$ by adding an addition measurement $\varepsilon_{p'} = \varepsilon_p \cup m_q$ where index $(m_i) >$ index (m_p), $m_q \in \overline{\mathrm{N}}_j$. That is, each measurement following from m_p in the ordered sequence of measurements and also in $\overline{\mathrm{N}}_j$ is in turn added to ε_p to form a new set of longer potential symptoms. For each $\varepsilon_{p'}$, recursively apply from step 3 setting $\varepsilon_{p'} = \varepsilon_p$ and setting k = j.

Only the simplest (shortest set of) symptoms are required and the depth-first strategy ensures all equally simple symptoms are located first thus terminating the search. Also contributing to the efficient selection is the characteristic that measurements contributing to the greatest number of remaining $\overline{\mathrm{N}}$ elements are always earlier in the sequence, and because these are the measurements that distinguish the failure state from the maximum number of nominal measurements, they are the best candidates to form completed symptoms using fewest measurements.

The ordering of the selection from the |N| sets making up $\overline{N}$ is arbitrary, but once an order is chosen, it determines the indices of the actual measurements, and must then be applied consistently. The reason for ordering measurements and then selecting sets of measurements in this way is to produce symptoms that are guaranteed to not be part of any set of nominal observations.

The lower part of Figure 4 illustrates the process. For OBS^M_1 in the leftmost column, two symptoms are generated with two measurements required for both. Two initial partial symptoms (b, {M}) and (d; {M}) are possible as the initial nodes in the search tree. The measurement with the highest rank of 6 is b, and provides two elements $\{\overline{N}_3; \overline{N}_2\}$ from $\overline{N}$, requiring only measurements from $\overline{N}_1$ to complete $\overline{N}$ and produce a symptom. The search from b is continued by traversing further down the measurements list for measurements involving $\overline{N}_1$. d is the next candidate and completes the set $\overline{N}$. The completed symptom is therefore ($b \wedge d$; {M}). It will be appreciated that variations to this sequence are possible, e.g. instead of comparing OBS_1 with OBS^M_1, OBS_1 could be compared with OBS_2 or OBS_3, etc.

For the second of the initial symptoms the next element of $\overline{N}$ required is $\overline{N}_2$. Considering lower ranking measurements in $\overline{N}_2$, f is the only possibility and also completes $\overline{N}$ for the symptom resulting in the symptom ($d \wedge f$, {M}). It will be noted that if f had not completed $\overline{N}$ the symptom would be abandoned without further searching since, if at all, it would provide a more complex symptom than ($b \wedge d$, {M}).

The second failure mode measurement OBS^M_2 produces $\overline{N}_1 = \emptyset$ and hence no symptoms are possible because all of the measurements are seen in $\overline{N}_1$ and manifests as an empty matrix column for $\overline{N}_1$ in Figure 4. OBS^M_3 produces a single symptom, although *a* would have been considered as a partial symptom but cannot be completed. Finally, the symptoms generated from each observation are added to an ordered tree of measurements to allow the complete set of faults related to each symptom to be captured.

Equivalent measurements

It is common for several different measurements to have equivalent diagnostic power within a symptom produced from a single fault observation. A simple concrete example is provided by two wires connected in series; a lack of current flow in either one will indicate the same faults. The presence of equivalent observations leads to more than one measurement with identical index(o) values in equation (15). These symptoms can be handled in two ways. Either symptoms must contain disjunctive expressions that allow for any one of a number of measurements to be used, or several independent symptoms are generated, one for each of the equivalent measurements. The symptom generation algorithm therefore simultaneously includes all measurements with the same index number as m_p at step 2 and 6.

Fault exoneration

The diagnostic symptoms as generated in the previous sections should not be negated to perform fault exoneration - a symptom expression evaluating to false does not necessarily indicate fault absence. For example consider a lamp with a plausible symptom:

S = (lamp $\leftrightarrow$ inactive $\wedge$ switch $\leftrightarrow$ on, {lamp blown, switch dirty, lamp wire fractured})

If the switch is off then switch $\leftrightarrow$ on is false resulting in a false symptom expression, but this does not imply the lamp is functioning correctly - it could be blown.

Manually crafted symptoms are often conditional so that ¬E will exonerate all of the faults in F. This is because engineers consider the implication of not seeing the symptom as well as its presence. The Bayesian net based diagnostic system for which the symptoms are being generated requires that symptoms can exonerate faults and therefore $(\forall M \in F, \forall n)(\neg E(OBS_n^M) \Rightarrow M \notin D)$. This also allows better fault detection by allowing evidence from nominally operating parts of a system to contradict evidence from general symptoms, hence reducing the set of possible faults.

Assigning $E_c \equiv$ switch $\leftrightarrow$ on and $E_o \equiv$ lamp $\leftrightarrow$ inactive, the earlier example can be reformed ($E_c \wedge E_o$, {lamp blown, switch contact dirty, …}) and ($\neg(E_c \rightarrow E_o)$m {¬lamp blown, ¬switch contact dirty, …}). The separate conditional part E_c must be satisfied for the symptom to be valid. Any valid symptom that is not satisfied (i.e. $E_c \wedge \neg E_o$) can be used to exonerate the associated faults. For example, if the switch is on and the lamp is active then we can predict that the lamp is not blown, the switch contact is not dirty, and the wire to the lamp is not fractured. A valid symptom that is satisfied implicates the associated faults. A symptom that is not valid provides no information.

To ensure negatable symptoms, E is partitioned to produce E_c and E_o to ensure the symptom is either satisfied or invalid for any observation OBS^M where $M \in F$. That is, a symptom expression must not exonerate a fault for an observation where the fault exists. Therefore if S = (E_c, E_o, F) measurements must be included in E_c to ensure:

$$(\forall M \in F, \forall n, \nexists OBS_n^M)(E_c \wedge \neg E_o) \qquad (16)$$

For symptoms formed as a conjunction of measurements, $\mathcal{E} = \{m_1, m_2, \ldots\}$ then $\neg E = \neg m_1 \vee \neg m_2 \ldots$ That is, if one of the measurements in the (proposed) symptom is not present in any of the observations for the fault(s) indicated by the symptom, then the symptom cannot be negated to exonerate the fault. To satisfy sentence (16) the symptom generator therefore partitions E for each symptom by finding candidate conditional measurements as follows: $\mathcal{E}_{c'} = \{o \in \mathcal{E} \mid \forall M \in F, \forall n : o \notin OBS_n^M\}$ Unfortunately, using all of these candidate conditional measurements is over cautious, and may lead to the inability of some symptoms to exonerate faults. Because $|\mathcal{E}_{c'}| \leq |E|$ and E is very small (see above), a simple depth first search rapidly finds all the smallest subsets of $\mathcal{E}_{c'}$ that satisfy proposition (16). Finally, $\mathcal{E}_o = \mathcal{E} \setminus \mathcal{E}_c$ is obtained.

Generalising Symptoms

This section details the use of functional information to produce generalised faults from an FMEA analysis which does not exercise all system states, or even all permutations of possible system functionality.

Measurements are associated with system functions based on the achievement and failure of functions across the entire FMEA. Given a comprehensive set of failures this provides a simulation derived description of which functions each component/measurement affects. This in turn allows an assessment of the relevance of measurements associated with a fault; measurements associated with functions that are never affected during the exercising of the fault (which should include at least all individual system functions) can be considered irrelevant. This technique ensures that irrelevant measurements are not included in symptoms, and hence that symptoms are as general as possible. Conversely, the decision as to which measurements are relevant is derived from the entire set of failure and non failure states encountered during the FMEA.

Figure 5 provides an illustration of the symptom generation for a system that provides two functions A and B, with two failure modes M_1 and M_2. Each of the nominal and failure modes produces a number of observed states related to the exercised functions in the scenario. The diagonal-hatched section indicates an observation used to generate a symptom for M_1. The cross-hatched sections represent states that are attainable in operation but not exercised by the FMEA for this example. The symptom generation described in section 3 deals with comparison of a failure state with the nominal operating states, and is shown by arrow labelled 1a. The thicker lines indicate comparisons that cannot be made because those states are not available. The symptom generation cannot consider the comparison shown by arrow 1b and therefore implicitly generalises the system, potentially generating a symptom that provides a false indication of fault M_1 in state OBS_{AB}. To resolve this issue there are two possibilities:

- Ensure the necessary states are provided by the FMEA. In practice by adding steps to the scenario longer spurious symptoms are generated that require more scenario steps, and ultimately a complete attainable envisionment is required. This approach is infeasible for most real systems.

- Constrain the search based on states related to structural and behavioural and functional interactions that have been encountered during the FMEA. This is achieved by associating component operation to function failure and also by associating component faults to affected functions on an FMEA wide basis. This information is then used at an individual observation level when selecting measurements from a failure observation for use a symptom.

Considering arrow 1b in Figure 5: given E are the symptom expression measurements of a symptom indicating a faultM_1 generated from an observation $OBS^{M1}{}_{A}$ when Ac(A), correct generalisation will be obtained if E is restricted such that if E were to indicate M_1 in unobserved state OBS_{AB}, then E must also indicate M_1 in state OBS_A:

$$\mathcal{E} \subseteq OBS_{AB} \Rightarrow \mathcal{E} \subseteq OBS_A \qquad (17)$$

There are two characteristics of the function model required to ensure proposition (17) holds:

- The allowable measurements E for $OBS^{M}{}_{A}$ must have some causal (structural or behavioural) relationship to the function A, and are not simply the result of `coincidental' consistency of measurement values due to limited observation of the system. Thus, excluding interactions with B, the fault causes function A to have the same behaviour with respect to E in both states $OBS^{M}{}_{A}$ and $OBS^{M}{}_{AB}$. This is discussed below under “Avoiding over generalisation”.
- E is compositional with respect to multiple functions that have not been exercised concurrently during the FMEA, i.e. the activation of B in addition to activation of A does not change the allowable

measurements E of the symptom for A. Essentially this means that the components or behaviours associated with the achievement of B must not be used in E for A unless this combination has been simulated.

The section “Associating measurements to functions” below develops a method to associate the components/measurements to functions based on observed behaviour and therefore allows those that interact to be excluded from E. This can be overcautious; however, and interacting functions that share components in their implementation may need to be exercised concurrently to allow sufficient measurements to be used to generate a suitably powerful diagnostic system. This is dependent on the engineers' expertise in scenario specification; functions that may have complex interactions or non-compositional behaviour elements should be exercised simultaneously in the FMEA. The diagnostic generator can provide information about excluded measurements, and the function combinations required to allow them to be used. This will both enhance the FMEA analysis and improve the symptom set.

Many systems contain components that are shared between many or all functions, for example the power supply. Fortunately, the effect of faults in these components is usually either the combination of effects on individual functions or the worst of the individual effects; in fact the automated FMEA actually relies on this characteristic to allow a compact scenario to be used. By assuming this level of fault-effect composition, it is often adequate to simply provide an “everything activated” state(s) in the scenario, and in more complex systems to simultaneously activate functions that share circuitry or components to prevent overcautious measurement selection for these faults. In the following sections measurement selection is developed that can select measurements when only some combinations of functions can be concurrently achieved.

Considering again Figure 5, arrows 2a and 2b represent comparison of observations between failure states M_1 and M_2. Comparison 2a (dashed-line arrow) is not part of the symptom generation described in section 3 and could not produce spurious symptoms if all possible symptoms were generated for each fault, since the same symptom expression would be generated for both faults if E was a subset of both observations. However, since minimal symptoms are being generated, there is the possibility that a symptom that would be generated for M_1 that also indicates M_2 may not be generated because a simpler symptom is available for M_2. It is desirable that symptoms are complete so that a valid symptom should indicate all detectable faults and therefore given S = (E; F) is a symptom and E(OBS) is the result of evaluating E on a set of observations:

$$(\forall OBS_n^M)((M \in F) \Leftrightarrow E(OBS_n^M)) \qquad (18)$$

This leads to an additional element in the symptom generation. Each completed symptom must be compared with the observations associated with all other failure mode observations and additional faults included in the symptom to ensure constraint 18 is satisfied. The unavailability of comparisons illustrated by arrow 2b provide the potential for a symptom to be generated that gives a false fault indication for M_1 if the system enters the AB state and has a fault M_2. It is similar to the situation indicated by 1b and is addressed by the same approach by ensuring:

$$\varepsilon \subseteq OBS_{AB}^M \Rightarrow \varepsilon \subseteq OBS_A^M \qquad (19)$$

Avoiding over generalisation

There are two parts to the process of restricting the measurements available for symptom generation based on the associated function states.

- Identify which measurements are behaviourally associated with each function. This is described in the “Associating measurements to function” section below.
- Select the available measurements for symptom generation based on the failure state of functions for each observation. This is described in the “Restricting symptom measurements” section below.

Clearly measurements referred to in the functional model trigger and effect expressions notated (T(fn) and E(fn)) are associated to the function; however, most measurements that might be used for diagnosis are not contained in these expressions. In addition those measurements used in triggers and effects are often not directly available to a diagnostic system, particularly an on-board system, where indirect measurements of the inputs and outputs are required.

The concept of a “function associated measurement” is used to indicate a measurement has some behavioural or structural “causal” relationship to a function based on evidence from the FMEA. The measurements associated with fn are denoted by the relations Ab(fn) and Nom(fn) that provide respectively, the abnormal and nominal measurements associated with a function. Often these will simply be different value measurements from the same sensor however this is not assumed.

Associating measurements to functions

Given a comprehensive component fault list, almost all diagnostically interesting measurements (with the exception of externally controlled triggers) will be affected by at least one of the failure modes of the system. By associating these abnormal measurements with functions that simultaneously fail we build up a mapping of function associated measurements. Even measurements structurally adjacent to external triggers will be affected by faults in the connecting components. For example switch.position $\leftrightarrow$ on is a trigger however the current flow though the switch.contact will be affected by faults such as the contact being stuck or wiring faults in the switch circuit and could be used as diagnostic measurements.

It is normal practice to exercise each function individually during an FMEA and therefore abnormal measurements are available for each isolated function. Computing Ab(fn) and Nom(fn) from an FMEA using the definition of RfnAb and RobsAb from equations (7) and (10) is a matter of considering all the observations in the FMEA :

$$\mathrm{Ab}(fn) = \bigcup_{\forall M, \forall n \leq |S|} \begin{cases} RobsAb_n^M, & \text{if } RfnFa_n^M \leftrightarrow \{fn\} \wedge RfnAc_n \leftrightarrow \emptyset \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\mathrm{Nom}(fn) = \bigcup_{\forall M, \forall n \leq |S|} \begin{cases} OBS_n \backslash (OBS_n^M \backslash RobsAb_n^M), & \text{if } RfnFa_n^M \leftrightarrow \{fn\} \wedge RfnAc_n \leftrightarrow \\ \emptyset, & \text{otherwise} \end{cases}$$

(20)

Generally, this produces a set of measurements that agree with engineering expectation of the components used to implement each function. However, even if this is not clear, there must be a structural or behavioural relationship between the function and the measurement, as required for measurement selection. It will be noted that Ab(fn) may include measurements that occur in observations when Ab(fn) is not failed. Membership of Ab(fn) implies only that a measurement can be affected by the failure and not that it will be in all states. The same applies to Nom(fn) and in general Ab(fn) $\cap$ Nom(fn) $\neq \emptyset$.

<u>Restricting symptom measurements</u>

Several variations of measurement selection are possible that trade the simplicity of measurement selection against the power of the symptom and the possibility of spurious symptoms. All variations redefine the measurement selection s in equation (12) to provide a restricted set of measurements. The initial approach selects all abnormal measurements plus triggers associated with the failed functions (defined in equations (7)-(9)) for the observation.

$$s = \{(RobsAb_n^M\} \cup \mathrm{T}(fn)) \mid fn \in RfnFa_n^M\} \cap OBS_n^M \qquad (22)$$

Experiments have shown that where operating modes combining multiple functions were deliberately restricted in the scenario, together with experience from other systems, and have identified the need to be more selective and resulted in the selection of measurement based on function. Using the function associated measurements to provide tighter selection of the abnormal measurements, can give better protection against spurious symptom generation, when the final system will exercise function combinations not found in the FMEA scenario.

$$s = \{(Ab_n(fn) \mid fn \in RfnFa_n^M\} \cup T(fn)) \cap OBS_n^M \qquad (23)$$

Allowing nominal observations in addition to triggers will increase the range of symptoms that can be generated:

$$s = \{(Ab_n(fn) \cup \mathrm{Nom}_n(fn)) \mid fn \in RfnFa_n^M\} \cup T(fn)) \cap OBS_n^M \qquad (24)$$

In particular, for systems where a fault only causes artefacts in part of the affected function(s) behaviour, it allows symptoms to be generated that identify the internal inconsistency in the function. For example, if one branch of a parallel circuit (providing a single function) fails then symptoms are generated that measure nominal flow in one branch and abnormal in another. It is thus important that the major sub-states of the function are exercised, such as any available control of the branches of a parallel circuit, particularly if `fully instrumented' systems are being investigated as a prelude to sensor selection.

Fully instrumented systems make large numbers of observations available for symptom generation, and many alternative symptoms can be generated representing alternative combinations of internal measurement inconsistencies. An engineer can then use external information and diagnosability tools to select the most convenient measurements to achieve the required diagnosability, for example.

Measurements selected using equations (23) and (24) can still lead to spurious diagnoses for novel functional combinations that share components due to false exoneration as illustrated in Figure 6. Assume the functions A and B were not exercised simultaneously during the FMEA. The symptom correctly predicts the fault (w2 fracture) when function A only is active but actually exonerates the fault when both functions are triggered and Ac(B) and Fa(A) because w1 is active. One solution is to exercise both of the functions simultaneously in the scenario, forcing the symptom generation to include additional measurements in the symptom, for example the position of switch sw2. As described previously, for some systems it is possible to activate all of the system functions simultaneously in the FMEA scenario; however, complex systems often have limitations on the combinations of attainable functions precluding this approach. A “safe” solution is therefore to disallow measurements associated with several functions if any of the functions is In or Un.

If ASh is defined as the abnormal measurements that are not shared with inactive function combinations and given fn $\in$ RfnFa_n^M and fs $|$ (RfnIn_n^M } $\cup$ RfnUn_n^M), the measurement selection in (23) becomes:

$$ASh = \{\, x \mid x \in \mathrm{Ab}_n(fn) \wedge x \notin \mathrm{Ab}_n(fs) \,\}$$

$$s = (\{x \in ASh \mid fn \in RfnFa_n^M\} \cup \mathrm{T}(fn)) \cap OBS_n^M \qquad (25)$$

Similarly, for the nominal measurements NSh, the measurement selection in (24) is modified to:

$$NSh = \{\, x \mid x \in \mathrm{Nom}_n(fn) \wedge x \notin \mathrm{Nom}_n(fs) \,\}$$

$$s = (\{x \in (ASh \cup NSh) \mid fn \in RfnFa_n^M\} \cup \mathrm{T}(fn)) \cap OBS_n^M \qquad (26)$$

This approach can be overcautious leading to a reduction in the number of diagnosable faults and particularly the number of operating states where faults are diagnosable. If a significant difference in the diagnosability of the system exists between equations (26) and (24) the function combinations causing measurements to be abandoned based on function sharing can be reported and the scenario can then be extended. Alternatively, in some situations, for example where functions are mutually exclusive (activated by different positions of a switch), they can be included in a list of allowable unexercised function combinations for shared measurements. There may be scope for determining mutually exclusive functions from the function model, but this has not been investigated because in practice the issue has not arisen.

Neither of the measurement selection mechanisms above make any use of unexpected function achievement to produce symptoms. This is for two reasons.

- Symptoms based on unexpected functions would require the absence of trigger measurements in an observation to be included as part of the symptom expression making it necessary to identify the sets of mutually exclusive measurements that form ¬T(fn) to allow suitable measurements to be selected that represent the trigger absence.

- Since Nom(fn) are required for Un(fn), it is necessary to ensure that all abnormal (failed and unexpected) functions present in an observation also exist in their achieved form in the nominal observations. Failing to do this would allow symptoms based on nominal measurements of function combinations that have not been observed in the nominal state.

Diagnosis framework

The target application to can use the generated symptoms in a Bayesian network to produce final diagnoses:

- The component fault prior probability describes component reliability.
- The symptom leak probability captures the probability that the symptom is observed even though there is actually no fault.
- Fault symptom conditional probability captures the probability that when the symptom is observed it is due to the fault indicated rather than any other fault. This can be useful for noisy signals, for example.

The probabilities allow for intermittent or unreliable measurements to be used. Knowledge of expected measurements is used to exonerate components from detected faults, as well as using knowledge of faulty measurements to implicate components in detected faults.

Multiple faults

The diagnostic system can only explicitly diagnose multiple faults if a multiple fault FMEA is performed. A multiple fault FMEA can be efficiently performed using this technology. Multiple fault FMEA analysis can by run for example by selecting only combinations of faults with higher combined component failure likelihoods. The multiple faults are then considered as a single compound fault within symptoms. If f_{12} represents simultaneous occurrence of two failures $f_1 \in$ M and $f_2 \in$ M, the symptom set could be modified by multiple fault FMEA in one or more of the following ways:

1. A unique symptom, S_{NEW} = (E, $\{f_{12}\}$) is produced that would allow the multiple fault to be isolated.

2. A symptom, S_n = (E, $\{f_3; f_{12}\}$), is produced that is identical to the symptom for a different fault.

3. Symptoms are produced that are the same as some or all of the individual faults. S_n = (E, $\{f_1; f_{12}\}$) or S_n = (E, $\{f_2; f_{12}\}$) or S_n = (E, $\{f_1; f_2; f_{12}\}$) .

4. The multiple fault case produces no symptoms.

Without using multiple fault FMEA, it has been found empirically for the systems tested, the most common effect of multiple faults is that all or some of the individual faults appear in the highest rank of possible faults (case 3 above). Where the faults are associated with the same function it is common for one fault to mask others, and the more pervasive fault appears in the highest ranking set. For case 1 and 4 above the multiple fault is not diagnosed. If only case 4 occurs then the multiple fault is not diagnosable, this could be two faults that negate each other and provide no overall effect on the system.

Case 2 is the only one where the multiple fault could lead to a spurious diagnosis; f_3 would be diagnosed if f_1 and f_2 occurred simultaneously. Empirically, it has been found that the most likely situation is a case of fault masking where all the faults are associated to the same function and f_3 masks the multiple fault. Parsimonious principles dictate that the diagnosis f_3 would be a preferred initial diagnosis, and is the result provided by the symptoms produced without multiple faults. Once f_3 is exonerated, by exercising additional faults or including additional measurements, without the multiple fault simulation the fault could not be diagnosed. For example, if two lamps wired in parallel both fail: the effect is the same as the main supply failing, and given that it is impossible to distinguish these faults the single fault is more likely unless the supply can be exonerated.

The system and technique described herein began with an investigation into replacing the use of manual FMEA as the input to a Bayesian network based diagnostic system with an automated FMEA. It became clear to the inventors that the comprehensive nature of an automated FMEA actually makes the identification of general symptoms more time consuming, although the resulting diagnostic system is more powerful. Automation of the generation of the symptoms was then considered, although emulating the selectivity and additional knowledge imparted by an engineer proved to be a challenge, and was addressed by the use of the functional model that already plays the role of interpreting detailed behaviour into a more meaningful abstracted description. An unexpected possibility was the ability to generate symptoms based on all potentially measurable quantities in a system. The resulting large set of symptoms (1000+ for typical automotive system examples) paves the way for a sensor selection tool that allows an engineer to quickly analyse which measurements may be most useful for a given set of "diagnosability" criteria. The potential diagnosability of a system can therefore be investigated in broad qualitative terms early in the design, possibly leading to alternative solutions that build diagnosis into preliminary design rather than as a retrofit activity. An early version of this tool is documented

There are several advantages of this type of system/appraoch described herein compared to an on-board simulation based system:

- The diagnostics can be checked and verified prior to deployment, which is useful where stringent certification requirements exist.
- The simulation effort need be carried out only once, and is part of existing design analysis requirements, and does not need to be done on the on-board hardware.
- Additional symptoms can be inserted manually, based on experience if necessary. This allows symptoms based on specialist sensors outside of the modelling domain of the system to be included such as a temperature sensor measuring electric component temperatures. In addition outputs of components or subsystems that include their own black box diagnostic capabilities that cannot be modelled can be included in the diagnostic system. Finally, additional symptoms can be

included (or possibly excluded) where for pragmatic reasons modelling simplifications exist in the models used for the design analysis.

- If used in a Bayesian system, the weighting of symptoms can be adjusted if required to reflect any uncertainty regarding the applicability of a symptom, for example where measurements might be considered less reliable.
- The diagnostic system requires relatively simple on-board processing.
- The diagnostic systems capabilities can be characterised in advance and the sensing requirements and cost weighed against the diagnostic and fault isolation capability of the system. An engineer can experiment with sensor selection and assess the capability of the resulting diagnostic system.

Only faults that are included in the simulation (FMEA) can be diagnosed, and only some types of multiple fault can be diagnosed; however, this can be extended by performing multiple fault FMEA discussed above. Currently, symptoms are based on qualitative system states and do not include sequences of states in the description of behaviour that may indicate a fault; however, fault isolation may be carried out by considering symptoms related to multiple system states.

CLAIMS

1. A computer-implemented method of generating electronic symptom data for use in a diagnostic system (210), the method including:

i) obtaining (302) electronic Failure Modes and Effects Analysis (FMEA) data for a target system, the data including a plurality of observations, each said observation including a set of at least one measurement associated with a fault state or a nominal state of the target system;

ii) selecting (304, 306) a first said observation associated with a nominal state;

iii) selecting (304, 306) a second said observation associated with a fault state;

iv) identifying (306) at least one measurement in the second observation that is/are not present in the first observation, and

v) generating the electronic symptom data based on the identified at least one measurement.

2. A method according to claim 1, further including (prior to step ii)):

i') identifying said observations from the FMEA data associated with a fault state that indicates failure of a function of the target system, and

using these identified observations for the selection of the first observation and the second observation.

3. A method according to claim 1, including identifying said observations from the FMEA data associated with all said fault states that indicate failure of each said function of the target system.

4. A method according to claim 2 or 3, where, for each said second observation OBS^{M}_{x} related to a particular said failure mode M, defining an ordered set $\overline{N}$ comprising sets of said measurements of the second observation that are abnormal with respect to the measurements of the first observation.

5. A method according to claim 4, further including (following step iv)):

iv') assigning an index value to the measurements in the ordered set $\overline{\mathrm{N}}$, the index value (for a said measurement "o" being processed during an iteration "i" of the method) being defined by:

$$\mathrm{Index}(o) = \sum_{i=1..|\overline{N}|} \begin{cases} 2^{i-1}, & \text{if } o \in \overline{N}_i \\ 0, & \text{otherwise} \end{cases}$$

6. A method according to claim 5, wherein the steps ii), iii), iv) and iv') are repeated for several combinations of said observations from the Failure Modes and Effects Analysis data, resulting in a set of potential symptoms E_p, each comprising at least one measurement from every said set in the ordered set $\overline{\mathrm{N}}$, with each member of the potential symptom set E_p being associated with a said index value.

7. A method according to claim 6, wherein a total index value of each said member comprising a valid symptom is equal to 2^x, where x is a number of elements in the set $\overline{\mathrm{N}}$. .

8. A method according to claim 7, further including creating a search structure including the potential symptom set.

9. A method according to claim 8, wherein the search structure comprises a binary search tree structure, with a position of a said measurement of the potential symptom set in the search tree structure being based on a decomposition of the corresponding index value by a power of 2.

10. A method according to claim 9, wherein the step v) of generating the symptom data includes performing a search on the search tree structure to identify members of the potential symptom sets having a fewest number of said measurements.

11. A method according to claim 10, wherein the search comprises a depth-first search.

12. A method according to claim 1, wherein the step v) of generating the symptom data includes generating data relating to at least one said symptom

that, during a diagnostic operation, is detected by the at least one measurement.

13. A method of diagnosing a target system including:

inputting symptom data generated by a method according to any one of the preceding claims into a diagnostic system, and

using the symptom data in the diagnostic system to identify faults in the target system.

14. A computer program product comprising computer readable medium, having thereon computer program code means, when the program code is loaded, to make the computer execute a method according to any one of claims 1 to 12.

15. Apparatus configured to execute a method according to any one of claims 1 to 12.

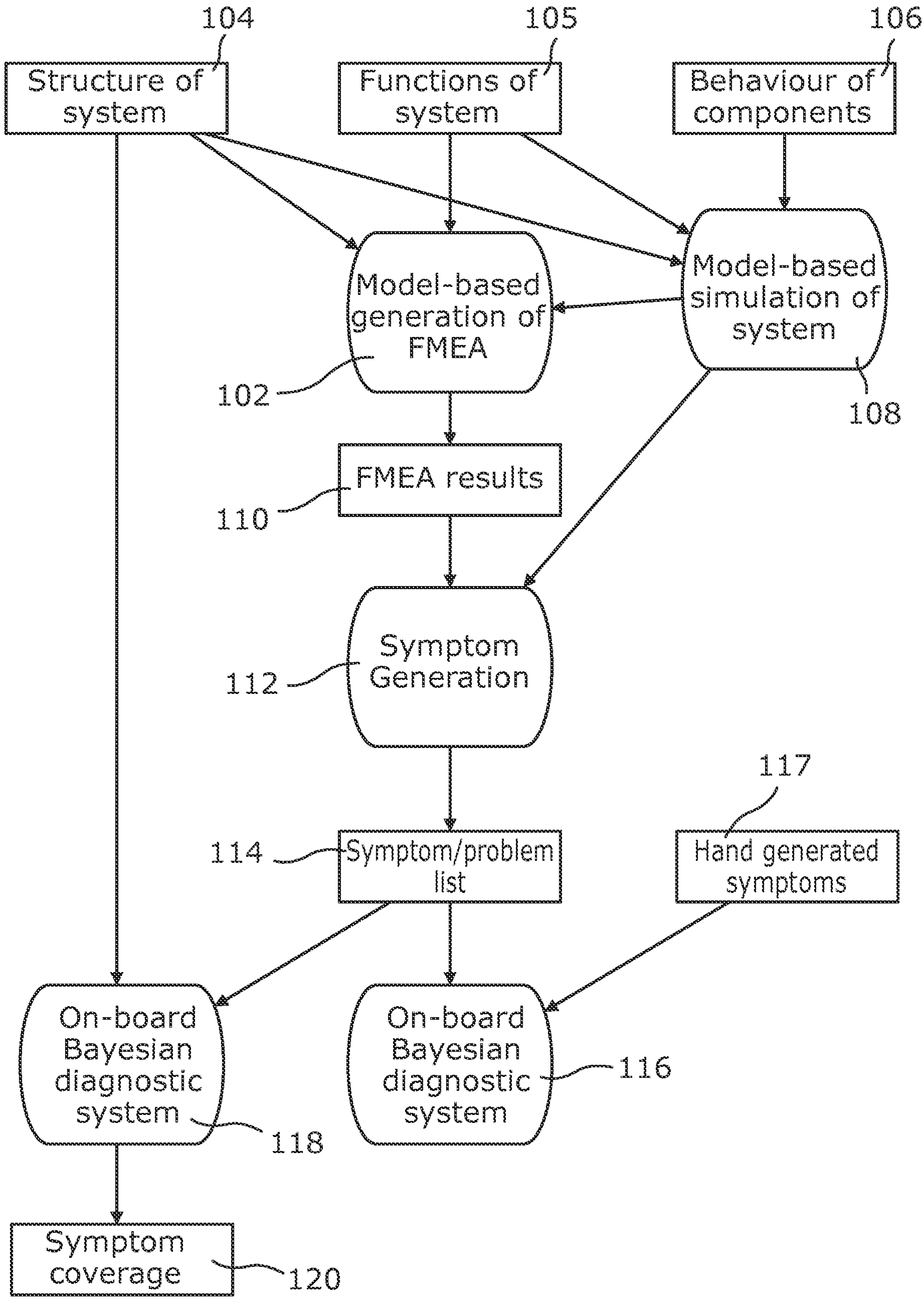
104
105
106
Structure of system
Functions of system
Behaviour of components
Model-based generation of FMEA
Model-based simulation of system
102
108
FMEA results
110
Symptom Generation
112
117
114
Symptom/problem list
Hand generated symptoms
On-board Bayesian diagnostic system
On-board Bayesian diagnostic system
116
118
Symptom coverage
120

Fig. 1

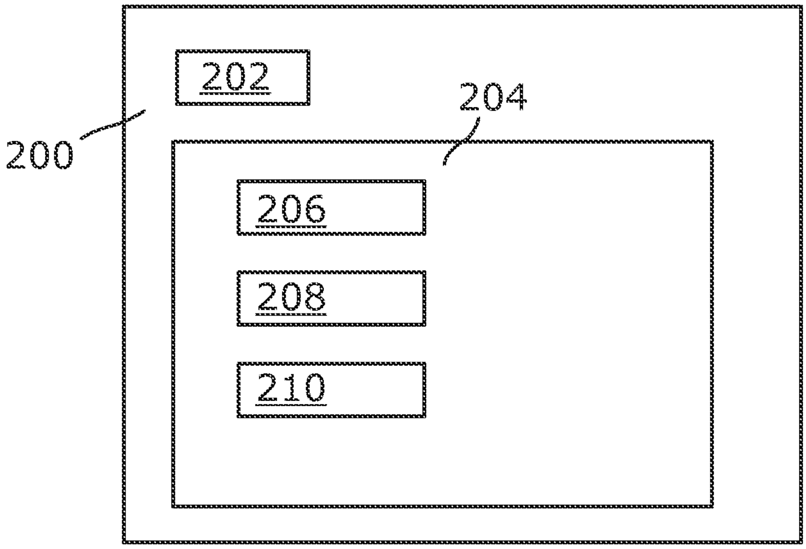
202
204
200
206
208
210

Fig. 2

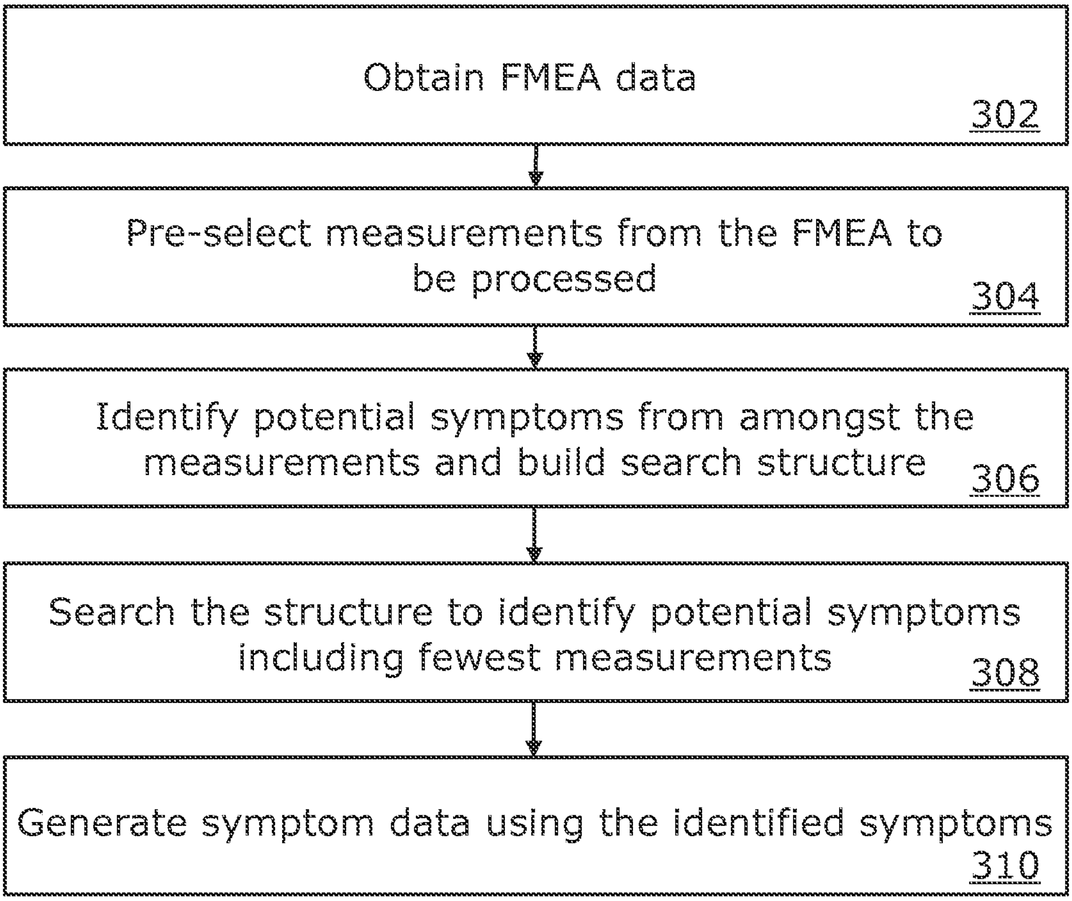
Obtain FMEA data
302
Pre-select measurements from the FMEA to be processed
304
Identify potential symptoms from amongst the measurements and build search structure
306
Search the structure to identify potential symptoms including fewest measurements
308
Generate symptom data using the identified symptoms
310

Fig. 3

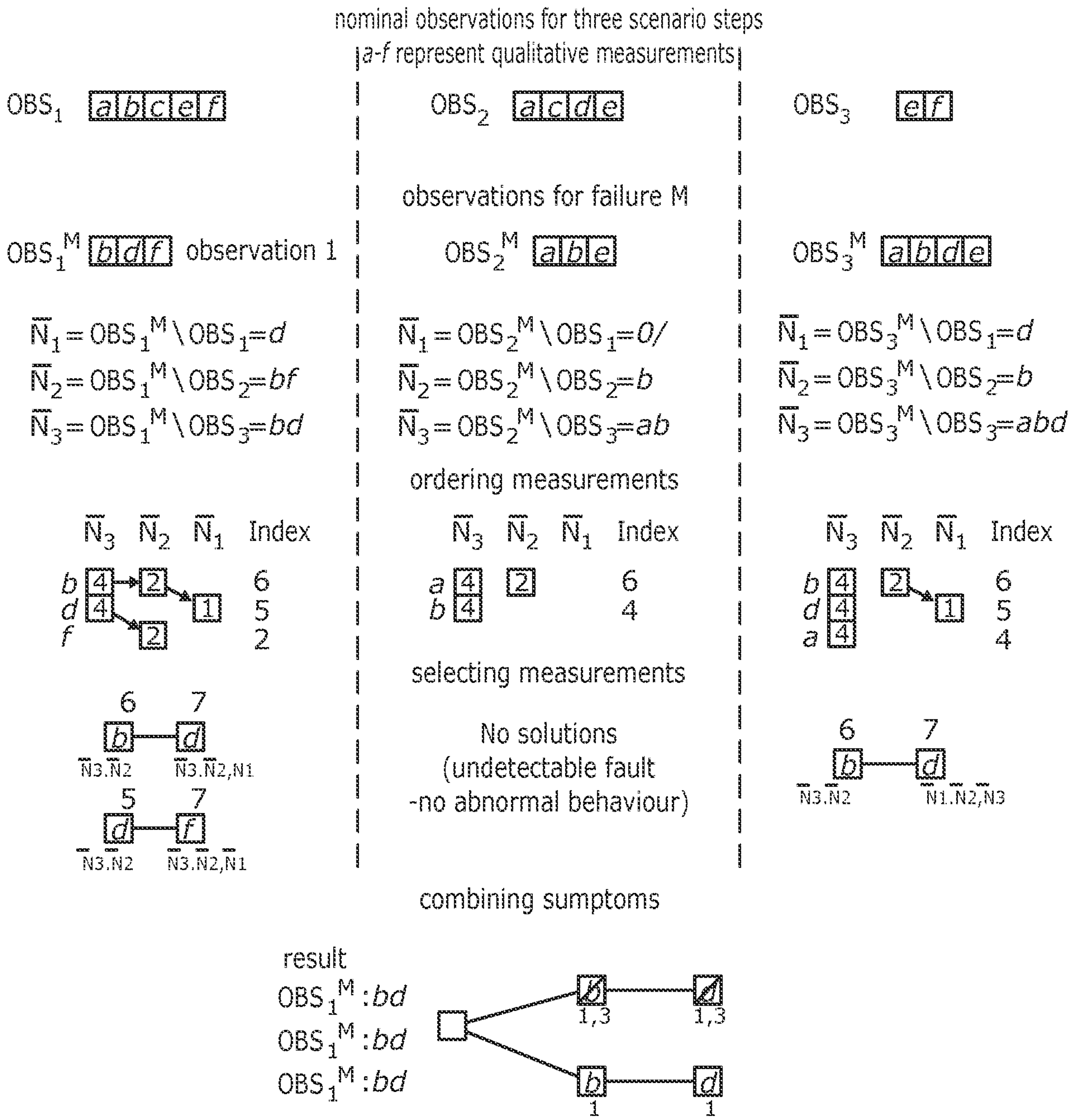
nominal observations for three scenario steps
a-f represent qualitative measurements
OBS1 a b c e f
OBS2 a c d e
OBS3 e f
observations for failure M
OBS1M b d f observation 1
OBS2M a b e
OBS3M a b d e
ordering measurements
Index
selecting measurements
No solutions
(undetectable fault
-no abnormal behaviour)
combining sumptoms
result
OBS1M : bd
OBS1M : bd
OBS1M : bd

Fig. 4

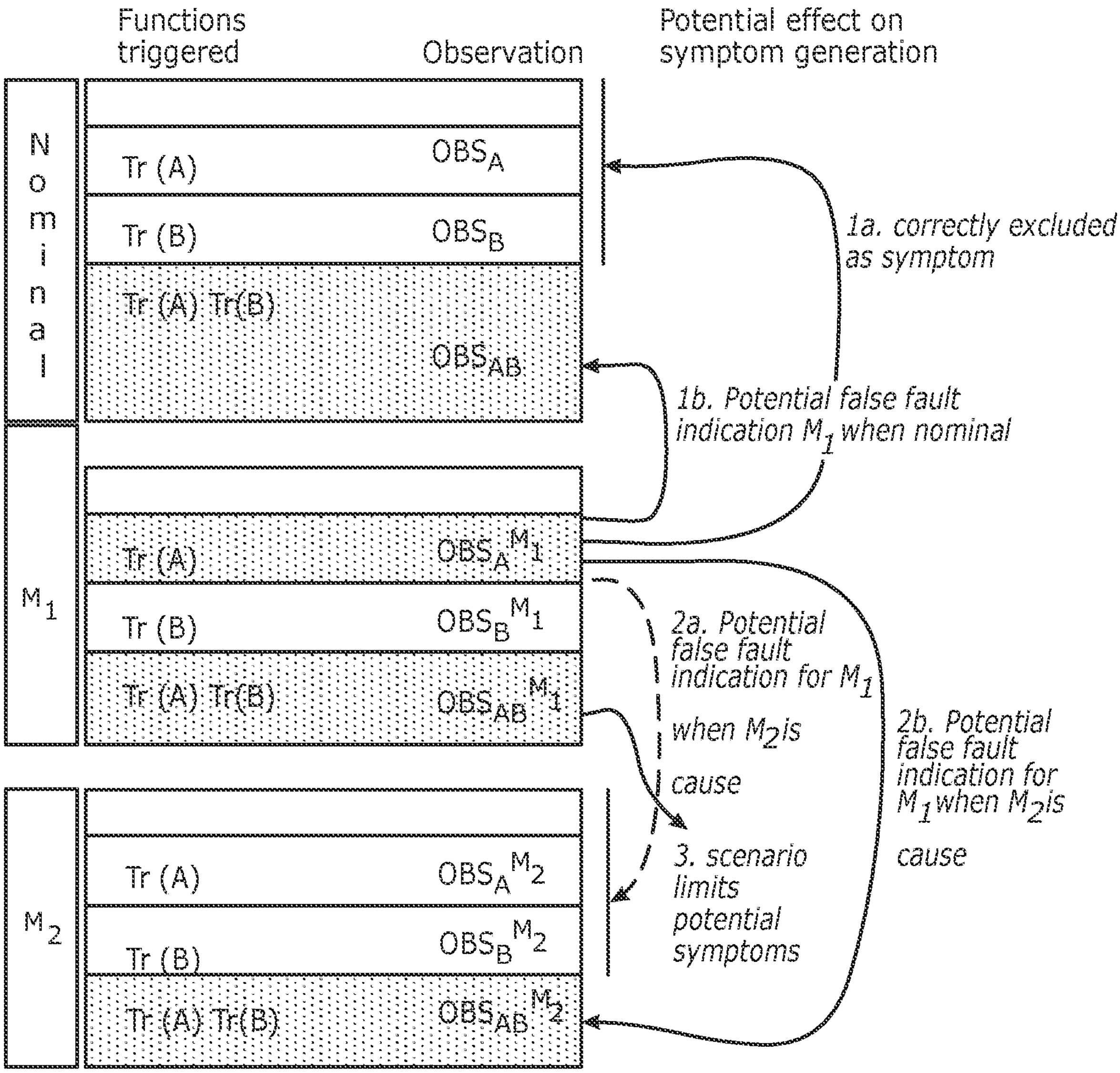
Functions triggered
Observation
Potential effect on symptom generation
Nominal
Tr (A)
OBS$_A$
Tr (B)
OBS$_B$
Tr (A) Tr(B)
OBS$_{AB}$
1a. correctly excluded as symptom
1b. Potential false fault indication M_1 when nominal
M_1
Tr (A)
OBS$_A^{M_1}$
Tr (B)
OBS$_B^{M_1}$
Tr (A) Tr(B)
OBS$_{AB}^{M_1}$
2a. Potential false fault indication for M_1 when M_2 is cause
2b. Potential false fault indication for M_1 when M_2 is cause
3. scenario limits potential symptoms
M_2
Tr (A)
OBS$_A^{M_2}$
Tr (B)
OBS$_B^{M_2}$
Tr (A) Tr(B)
OBS$_{AB}^{M_2}$

Fig. 5

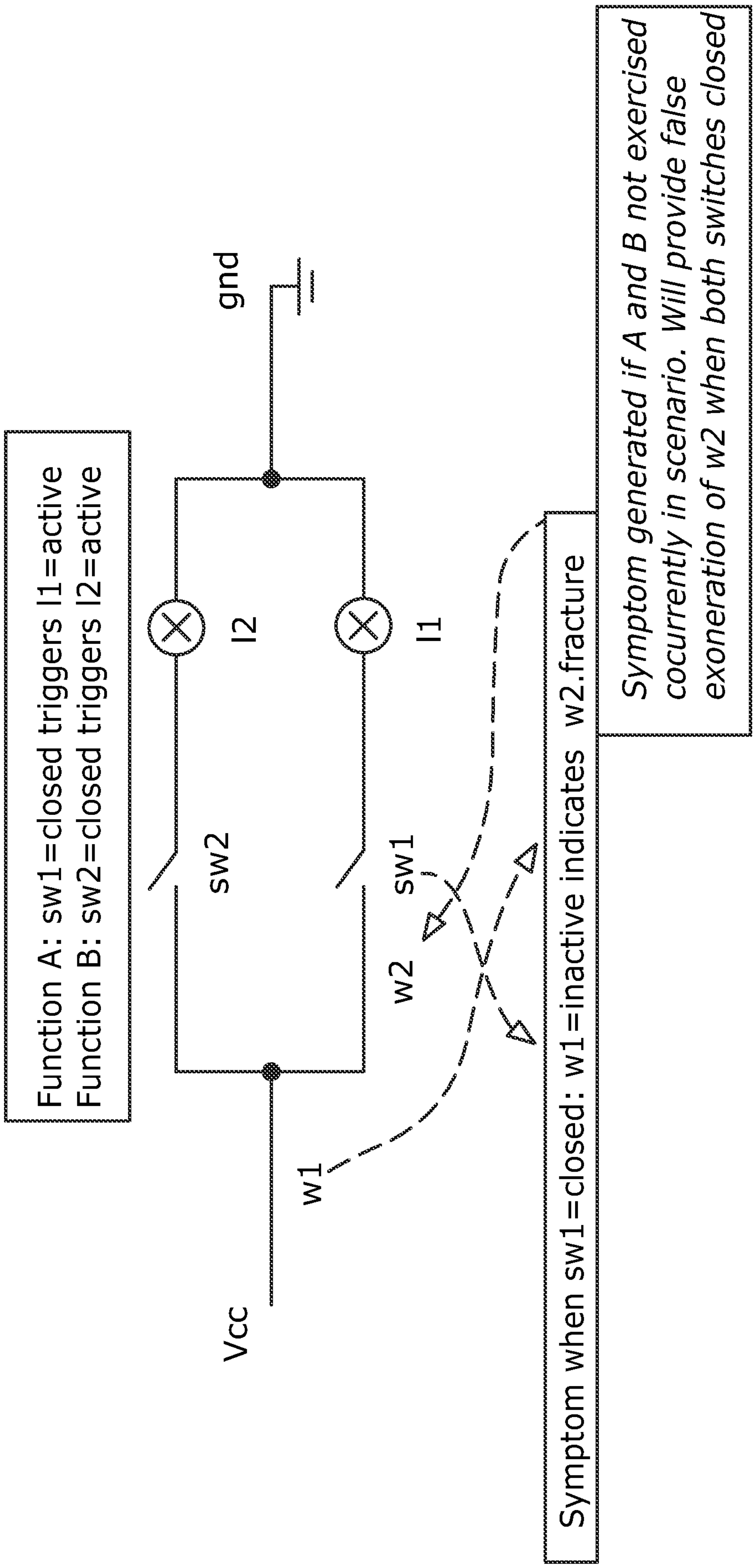
Function A: sw1=closed triggers l1=active
Function B: sw2=closed triggers l2=active
gnd
Vcc
l2
l1
sw2
sw1
w1
w2
Symptom when sw1=closed: w1=inactive indicates w2.fracture
Symptom generated if A and B not exercised cocurrently in scenario. Will provide false exoneration of w2 when both switches closed

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2012/050799

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/50 G05B23/02
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Neal Snooke: "Generating a diagnostic system from an automated FMEA", Proceedings, Annual Conference of the Prognostics and Health Management Society 2009San Diego, CA September 27 - October 1, 2009 , 27 September 2009 (2009-09-27), pages 1-12, XP002662194, Retrieved from the Internet: URL:http://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2009/phmc_09_60.pdf [retrieved on 2011-10-25] the whole document, in particular section 2 ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 May 2012	06/06/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Lerbinger, Klaus

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2012/050799

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2003/195675 A1 (FELKE TIMOTHY J [US] ET AL) 16 October 2003 (2003-10-16) the whole document -----	1-15
A	WO 2010/142977 A1 (BAE SYSTEMS PLC [GB]; SNOOKE NEAL [GB]) 16 December 2010 (2010-12-16) the whole document -----	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2012/050799

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003195675 A1	16-10-2003	AU 2003223569 A1	27-10-2003
		EP 1495384 A1	12-01-2005
		US 2003195675 A1	16-10-2003
		WO 03087967 A1	23-10-2003
WO 2010142977 A1	16-12-2010	EP 2440980 A1	18-04-2012
		US 2012131388 A1	24-05-2012
		WO 2010142977 A1	16-12-2010