Generating a diagnostic system from an automated FMEA

Neal Snooke [1]

[1] *Aberystwyth University, Department of Computer Science, Ceredigion, SY23 3DB, United Kingdom*
nns@aber.ac.uk

ABSTRACT

This paper builds on the ability to produce a comprehensive automated Failure Modes and Effects Analysis using qualitative model based reasoning techniques. From the FMEA output a diagnostic system comprised of a set of symptoms and associated potential faults can be generated and used as the basis of an on-board or off-board diagnostic system. This makes it is easy to propose additional sensing possibilities for the system, however a method is required to allow an appropriate set of sensors to be selected that provide the required level of diagnosability. The large number of competing factors outside of the scope of the modelling combined with the additional system knowledge required makes it difficult to optimise automatically. This paper therefore documents a semi automated technique that provides an engineer with easy access to information about diagnostic capability via a matrix visualisation technique. The focus of the project was the fuel system of an Uninhabited Aerial Vehicle(UAV) although the system has also been used on an automotive electrical system, and is applicable to a wide range of schematic and component based systems.

1 INTRODUCTION

This paper presents a technique to allow an engineer to investigate the relationship between sensor selection and the ability of a one step diagnostic system to detect faults. It has been developed as part of ASTRAEA(ASTRAEA, 2009), a pioneering £32 million UK aerospace programme which is addressing key technological and regulatory issues in order to open up non-segregated airspace to uninhabited autonomous aircraft.

Failure mode and effects analysis is a technique that is used to provide a comprehensive description of the effects of component faults. In the ASTRAEA project automated FMEA results were processed generate symptoms for an onboard diagnostic application. This automated technique allows for many possible additional sensors to be included, however pragmatically a subset of the possibilities must be chosen and the challenge is to decide which sensing provides for the the ‘best’ diagnostic system.

The symptoms can be generated manually from an FMEA or by automated simulation techniques (Price *et al.*, 2006; 1997). This work focusses on symptoms generated automatically, because the approach allows new diagnostic systems to be rapidly generated based different sensor sets or allow a fully instrumented system to be generated that allows any system parameter to be considered in the symptom generation process. The impact of design changes can also easily be incorporated into the diagnostic system. While manual FMEA generally only considers functional effects of failures, an automated FMEA is able to provide details of specific behaviour such as sensor readings. This allows a very comprehensive set of symptoms to be generated that includes observation combinations that may not be immediately obvious to an engineer and in any case would be very tedious to generate manually. The automated generation of symptoms from an FMEA will be the subject of a future paper.

In many applications there are various costs (financial, mass, layout, harness complexity etc) involved with each sensor resulting in a need to compromise between diagnostic ability and sensing. Typical issues that arise are:

- Which faults are diagnosable by the system?
- Which additional sensors could be included to diagnose additional or critical faults?
- What is the best ‘diagnostic value’ that can be obtained by adding additional sensors.

Due to the complexity of the mapping between sensors, symptoms and faults it is a non trivial task for an engineer to answer these questions without tool assistance. Many existing optimisation methods are either very specific solutions to an individual system e.g. (Maul *et al.*, 2007; Mushini and Simon, 2005), or generic and do not allow varied additional application

Ce	Oe	Faults indicated
false	false	$\emptyset$ (no fault information)
false	true	$\emptyset$
true	false	$\neg F$ ($\emptyset$ for non negatable symptoms)
true	true	F implicated

Table 1: Symptom states

specific information to be taken into account (Debouk *et al.*, 1999; Trave-Massuyes *et al.*, 2006). The problem has large search spaces and techniques such as genetic algorithms (GA) are often used to find solutions (Spanache *et al.*, 2004; Mushini and Simon, 2005; Maul *et al.*, 2007). We found empirically that in many cases there were simply too many additional considerations that an engineer can resolve but which would be difficult to provide to a fully automated system.

This work therefore focuses on assisting an engineer to use their knowledge, while retaining the comprehensive analysis that comes from a detailed whole system model-based simulation of failures.

The following sections of this paper firstly outline the generation of the symptoms and their characteristics and we briefly describe a software tool to allow an engineer to explore the diagnostic system using a simulator. A graphical approach is presented to assist an engineer to quickly visualize the diagnostic behavior of the system. (Thompson *et al.*, 1999) also describes a graphical tradeoff of competing requirements however these are aimed at architectural choices rather than sensor selection. This allows rapid investigation of the sensor selection and placement options available. The technique has been used on several case studies including an aircraft fuel system and an automotive Daylight Running Light (DTRL) electrical system and these systems with differing diagnostic characteristics are presented as case studies to illustrate the diagnostic system generation.

2 SYMPTOM GENERATION AND THE DIAGNOSTIC SYSTEM

Given a set of symptoms $S_1..S_N$ derived from an FMEA, each symptom is comprised of a tuple of (Ce, Oe, F) where both Ce and Oe are logical expressions and F is a non empty set of faults that are indicated when the symptom is satisfied. Each of these associated faults will have produced an abnormal set of observations in the FMEA that will lead to the symptom being satisfied. Ce specifies when the symptom is applicable and is termed the *symptom condition expression*. If Ce is false then the symptom is considered invalid and cannot be used. Oe is termed the *symptom expression*. If $Ce \wedge Oe$ evaluates true then one or more of the faults F are indicated. Table 1 shows the possible states of a symptom.

The third row illustrates a 'negatable' symptom able to exonerate faults ($\neg F$) and is the reason for Ce expressions. Negatable symptoms typically produce fewer symptoms but require more terms in the expressions than non-negatable symptoms. The ability to exonorate faults when observations are absent is important when the symptoms are used in some forms of on board diagnosis based on for example Bayesian networks. For garage based or workshop applications it may be useful to use the non-negatable version where the concept of symptom validity is not required and the $Ce \wedge Oe$ conjunction can used as a single expression directly indicating associated faults.

Both Ce and Oe are logical expressions formed from boolean *observations* and the usual logical operators. Observations may be formed from any available sensor reading, variable, state or system parameter that can be observed. Inputs (externally controlled values) are also considered as measurements and in fact the diagnostic system does not need to differentiate inputs and outputs during symptom generation or when in use, although observations that are required in the conditional part of a symptom often turn out to be inputs to satisfy the definition of a symptom. Most sensors produce *measurements* and a comparison operator is normally used to for an observation (e.g. pressure $<$ 5, or flow $\neq$ high). The use of a qualitative simulator (Price *et al.*, 2003; Lee and Ormsby, 1991; Lee, 2000; Snooke, 2007) makes it unnecessary to consider numerical values at the symptom generation stage since all measurements produced by the simulator are from qualitative quantity spaces for example 'high', 'zero' 'lower than expected' etc. Typical real symptom examples are shown in table 2. The example symptoms demonstrate qualitative analysis; in the final row we see that when the pump (CP) is on and a valve (TVL) is set, a low flow transducer (FT) observation indicates a possible blockage in two places. Clearly a flow meter will provide a numerical value and a decision must be made as to what constitutes a 'low' value. This could be done in a number of ways including a characterization of each system during manufacture, however the ASTRAEA project has simply chosen appropriate thresholds. For real systems found in the aerospace and automotive application areas we find there are typically of the order of hundreds of symptoms, diagnosing hundreds of possible faults.

An example automatically generated diagnostic system produced from an automated FMEA is illustrated in figure 2 for a twin engine aircraft fuel system 1. The tool in the figure allows an engineer to exercise a diagnostic system by inserting known faults in the top panel. The values determined by the simulation are immediately shown in middle section. The functions are derived from a functional model of the system that is used in the generation of the symptoms as well as to provide interpretation of the behaviour for presentation to an engineer in an FMEA output (Bell *et al.*, 2007; Bell and Snooke, 2004; Snooke and Bell, 2002). Functions are not used in the evaluation of the symptoms (but do have a role in their generation) and are only shown in the interface to allow easy recognition of the

Ce	Oe	F
TVL_RL_LH.position==‘isolation’	TVL_FL_LH.tellback==‘crossover’	TVL_FL_LH.stuck_crossover
TVL_RL_LH.position==‘crossover’ ∧ CP_FL_LH.control==‘on’	OC_WT_RH.tank_level==‘higher than expected’	TP4_FL_LH.fracture TP2_FL_LH.fracture TP4_FL_LH.partialblocked
CP_FL_RH.Control==‘on’ ∧ TVL_RL_RH.position==‘normal’	FT_FL_RH.flow==‘low’	FL1_1_FS_RH.partialblocked TP5_FL_RH.partialblocked

Table 2: Example symptoms

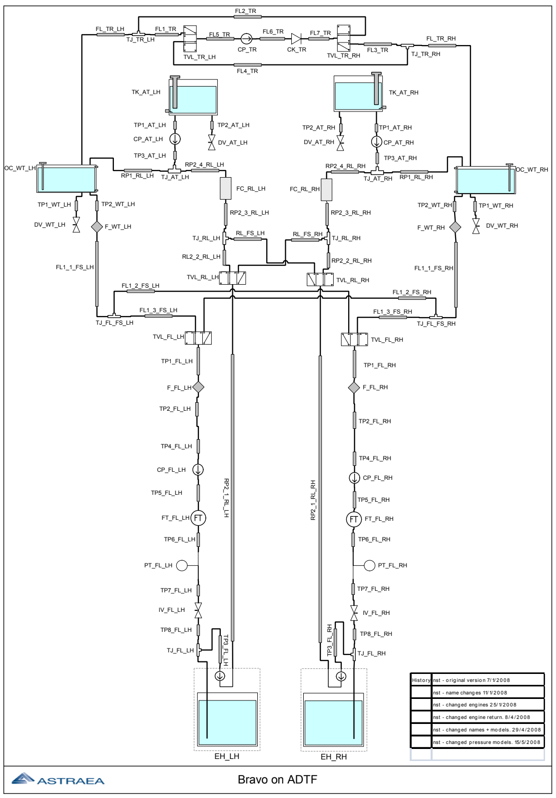

Figure 1: Fuel system schematic

overall effect of the fault to the user. The lower part of the screen shows the results of the diagnosis. On the left are all symptoms where $Ce = \text{true}$. The symptom set is negatable and therefore a check in the I/E column of figure 2 indicates that $Oe = \text{true}$ for the symptom and therefore indicates a set of faults. There is no check in the I/E column if $Oe = \text{false}$ and in this case the symptom will exonorate associated faults. A simple ranking of faults is provided based on the sum of the total number of symptoms indicating and exonerating each fault (shown in parenthesis). In this example there are 9 top ranking faults and these are in fact indistinguishable from the sensing available. The real diagnostic system includes other information about symptom and measurement confidence, using Bayesian methods to provide more fine grained fault ranking. This tool simply allows the symptom generation to be verified. Further down the list faults may have negative scores, showing that there is evidence from the symptoms that those faults are not present.

The engineer can select or deselect any sensor and the effect on the diagnosis is shown instantly and this is useful to check the applicability of specific measurements in specific fault scenarios, however it is not sufficient to allow an engineer to make a sensor selection for the system due to the number of possible operating modes and faults possible. It is this issue that provides the main focus of the remainder of this paper.

3 FAULT MATRICES

The relationship between observations (sensor measurements), symptoms and faults can be represented using two 2 dimensional matrices as shown in figure 3. A colour coding system is used to indicate the status of each element. Green indicates that an item is available to the diagnostic system (also a small tick is shown for clarity). Once a measurement is made available any symptoms that have all the necessary information to evaluate their Ce and Oe expressions also turn green together with any faults that can be diagnosed. A lighter green colour (centre dot) indicates that a measurement is available to a symptom but the symptom requires further measurements. If a measurement is to be excluded then it will be coloured red (a small cross shown) and any symptoms and faults that therefore cannot be diagnosed also turn red. Notice that it is necessary for all symptoms that can diagnose a fault to be excluded before the fault is not diagnosable.

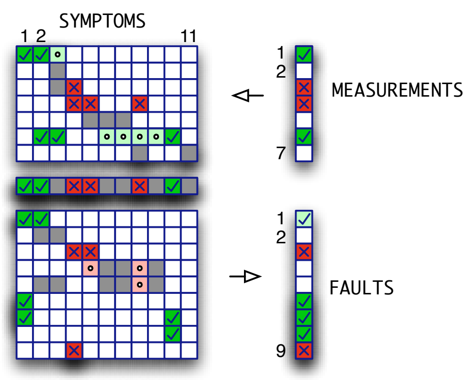

Figure 3: Measurement - Fault Matrix

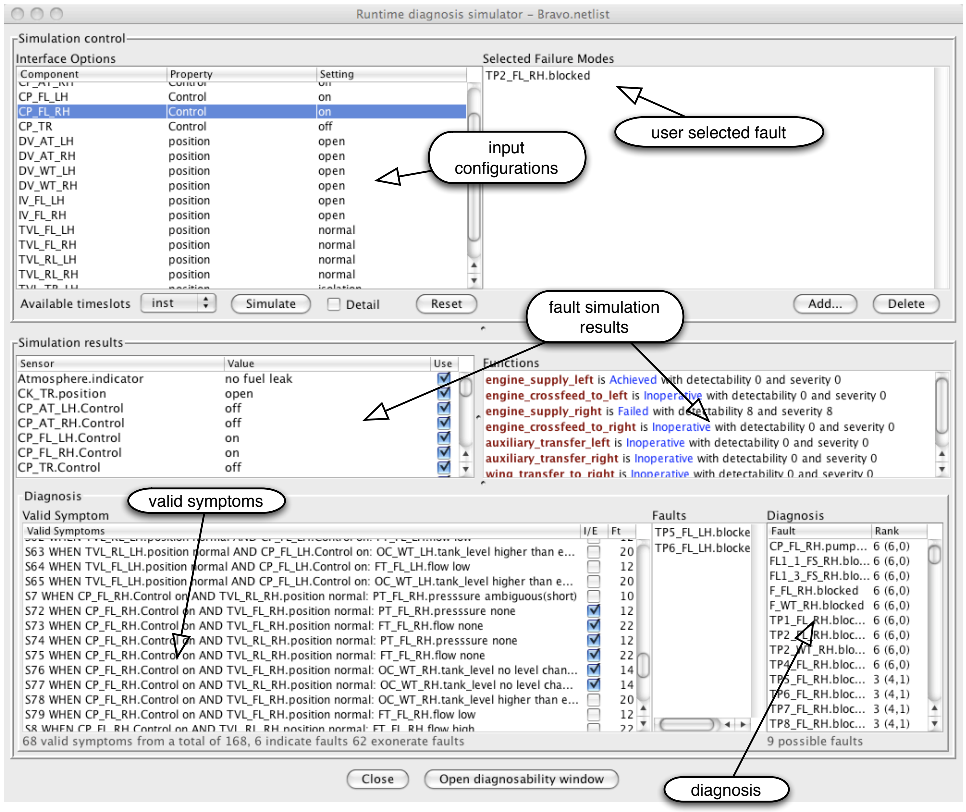

Figure 2: Diagnostic evaluator interface

Hence, cells that are pink (dot) indicate a symptom that cannot be used for a fault that can be diagnosed using an alternative. Elements that are undecided are coloured grey. This will be measurements that are neither chosen or excluded, symptoms that require undecided measurements and do not include excluded measurements, and faults that could still be diagnosed if additional symptoms (measurements) are included.

A real example for the aircraft fuel system is shown in figure 4 where it is clear that a structure exists in the fault behaviour of the system. Simply by selecting and deselecting measurements at any point in the measurement selection process it is easy to find out which (additional) measurements are significant in the context of the currently available measurements. In this figure the user has already selected some measurements and the result of this in terms of the symptoms and faults that can be diagnosed is shown in green (darker). The user is considering additional measurements and these are shown in yellow (lightest).

Patterns in the matrices graphically illustrate some characteristics of the diagnostic system:

- Highly populated rows in the measurement-symptom matrix shows measurements that participate in many symptoms and are therefore important to the diagnostic system.
- Similar patterns existing in more than one row of the measurement-symptom matrix indicate that there are several measurements required as a set, for a given a set of symptoms
- Highly populated columns in the measurement-symptom matrix indicate symptoms that require many measurements. In practice we find inputs such as valve positions and switches that affect major system state typically have this characteristic.
- Highly populated columns in the fault - symptom matrix indicate symptoms that can diagnose many faults.
- Similar patterns in several fault - symptom columns show that there may be a choice of symptoms that diagnose the same set of faults

3.1 The diagonal matrix

To gain an understanding of the relationships contained within the matrices an ‘approximate diagonal form’ can be generated for either matrix which attempts to place all the matrix elements as close to an imaginary line from top-left to bottom-right as possible. Since the matrices are not generally square a true diagonal matrix in the mathematical sense is not possible.

The concept of a row (or column) weight is used to describe the number of cells in either a row or column

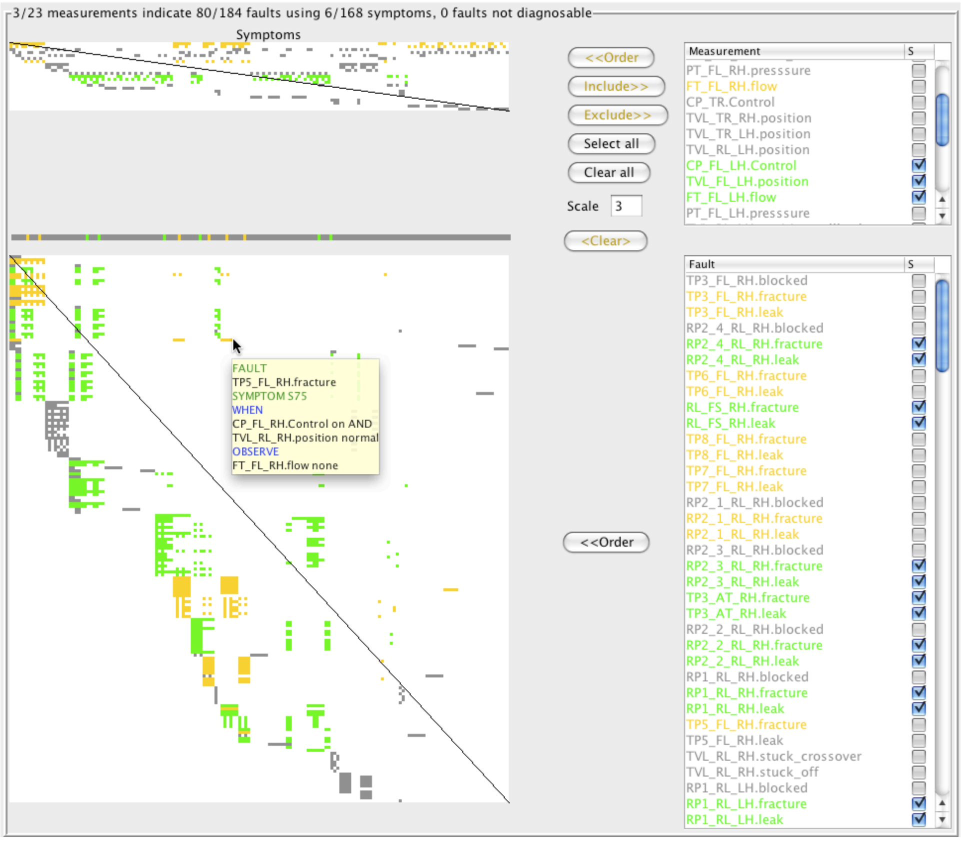

Figure 4: Aircraft fuel system matrix example

to either side of the imaginary diagonal line across the matrix. Figure 5 shows an example 6 by 4 matrix. The mid point of rows 1 and 2 are shown by the filled symbols. The weight of each row is calculated as the sum of the distance (as a cell count) of each active cell (shown grey in figure 5) from the mid point. In the upper matrix of the example row 1 has a weight of $\frac{2}{3}$ and row 2 has a weight of $-\frac{11}{3}$. By extension, the columns can be similarly considered. If the imbalance of two rows is defined as the weight of row $n-$the weight of row $n+1$, then the rows are swapped if the imbalance is greater than zero unless the result of swapping the rows creates a larger imbalance for the rows. In the example the imbalance is $\frac{2}{3} - (-\frac{11}{3}) = \frac{13}{3}$. This is greater than zero and therefore the rows are swapped to produce the matrix shown in the lower part of figure 5, in which the imbalance is $-\frac{1}{3} - (-\frac{9}{3}) = \frac{8}{3}$. Since $\frac{8}{3}$ is less than $\frac{11}{3}$ the reordered matrix is considered closer to diagonal than the original and the swap is retained. A similar procedure is then carried out between rows 2 and 3, and so on. The overall effect of swaps is to reorder the lists of measurements, symptoms, and faults. Each pair of rows are repeatedly considered in the manner of the known bubble sort algorithm, using the weight measure as the ordering criterion. However, in contrast to a standard sort the weight of a row changes (and is therefore recalculated) when it is moved. The sort is undertaken alternately on rows and columns.

Once each pair of row and column sorts is completed the total imbalance of the entire matrix is calculated as the imbalance sum of all rows plus the imbalance sum of all columns. The alternate sorting of rows and columns continues until no further reduction in the total matrix imbalance can be achieved. Once the chosen matrix is in diagonal form the unshared axis of the other matrix is sorted to make it as diagonal as possible. At this point the majority of the weight of the matrix is balanced around the diagonal as closely as possible. This has the effect of bringing related measurements and symptoms (or symptoms and faults) together on the diagonal and allows the user/engineer further insight to the diagnostic capability of the system.

The aim is to assist in the selection or removal of measurement and therefore any elements that are already decided are NOT included in the process and are moved to the bottom or right of the matrix. This is why the diagonal line does not extend the full size of the matrix in figure 12 (discussed later) which is also an example of a diagonal symptom-fault matrix showing distinct sets of symptoms that diagnose dis-

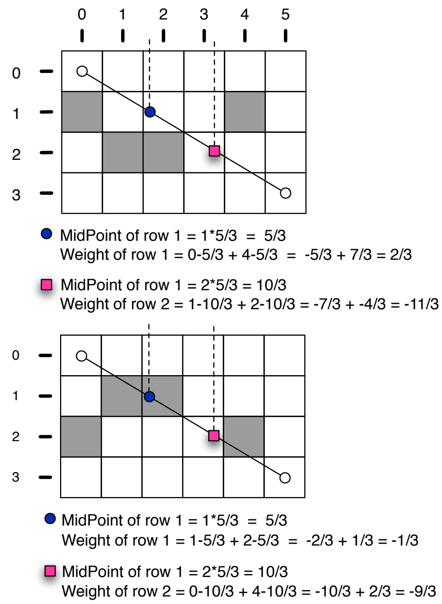

Figure 5: Producing the diagonal matrix

tinct sets of faults for an automotive system. In figure 6 the measurement matrix for the fuel system is diagonal (compare with figure 4) and in figure 11 the measurement matrix for the lighting system (discussed later) has been made diagonal and can be compared with figure 12 where the fault matrix is diagonal.

4 SENSOR SELECTION

The tool can calculate the maximum number of faults that can be diagnosed for up to n additional measurements dependent upon the system and time available. This is an exhaustive search and n can usually only be a small number since if r is the number of unselected measurements remaining there are $\frac{r!}{(n*(r!-n))}$ combinations of measurements to consider. The best solution may not necessarily be included in best solutions for larger numbers of measurements so hill climbing solutions do not work in general. In addition most non trivial systems have many possible 'best' combinations of n sensors, often due to symmetry in designs or sensors equivalent for some diagnostic aspect. An engineer is able to notice these features and decide that either all or none of a set should be included. For example there is little point in being able to diagnose a left hand circuit aircraft fuel system fault and not an equivalent right circuit fault. Of course once an engineer has identified a few critical measurements and perhaps excluded some unobtainable measurements r decreases allowing larger n to be considered.

4.1 The sensor selection advisor

To assist the engineer the tool can search for the next best n measurements in terms of the number of faults able to be diagnosed. An iterative process of measurement inclusion and exclusion can therefore be carried out until the a diagnostic system is produced with the required balance between measurement cost, symptom count (generally this is not a problem) and fault identification. The next section describes by means of an example how information is presented to the engineer at each iteration of the process.

For the example systems (with no previous selections) a search for $n = 3$ measurements takes seconds, 5 measurements a minute or so, with a maximum of 8-10 if an hour can be spared[1]. However higher numbers do not actually help in the measurement selection process, because the number of numerically equivalent solutions becomes overwhelming. The results naturally fall into a hierarchy:

1. The results are firstly grouped according to the number $m <= n$ of measurements involved and the number of diagnosable faults .
2. All of the measurements involved in any m measurement solution are listed, with any that also participate in a shorter (i.e. fewer measurements) solution indicated in a lighter font (e.g. CP_FL_LH.control).
3. The possible measurement combinations are grouped according to the fault set diagnosable.
4. The measurements required for each fault set is given. It is common to find several alternative sets of measurements that can diagnose the same set of faults and often the reason is obvious to the system engineer, for example in figure 9 we see that either the flow (FL) or return (RL) valve position can be used.

For clarification, an example of the hierarchy can be seen in a subsequent example at the lower right of figure 8.

5 EXAMPLE

The benefit of the diagnosability matrices and are best illustrated by a worked example of how an engineer might use the information to select a set of sensors and generate a diagnostic system. Consider the aircraft fuel system example of figure 4. From figure 6, it will be apparent to the skilled user/engineer that for this system most measurements are needed in several symptoms because of the horizontal bars in the matrices. If the user/engineer knows that the measurements from the flow meters are definitely available to the diagnostic system, then this can be selected in the measurement list by checking boxes as shown, resulting in the appropriate cells in the matrices turning green. However, it can be seen on the fault matrix that no cells turn green demonstrating that these measurements alone are not enough to diagnose any fault (see also the summary at the top of the window).

The pump control values are also known and can be selected, in figure 7. It can then be seen that these

[1] using a 2.4Ghz dual core Intel based laptop computer

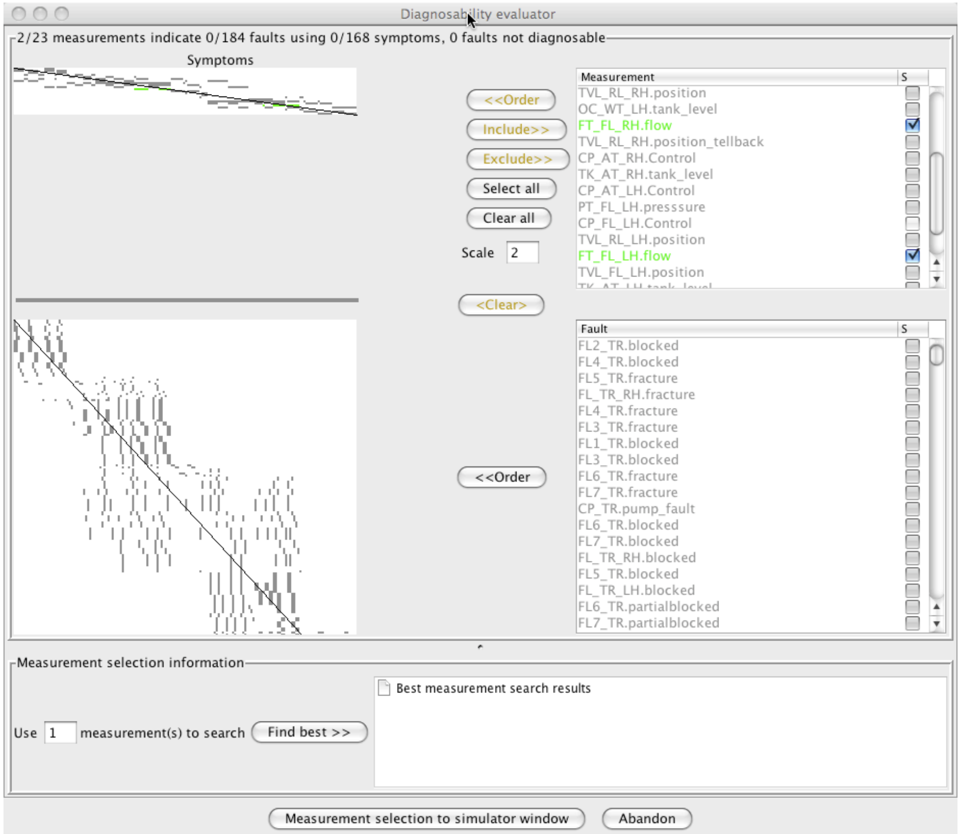

Figure 6: Fuel system - Selected flow measurements

observations are part of a symptom superset of the flow values and so the user may appreciate that it might be better to use them as a starting point instead of the flow meters.

The flow meter measurements could be deselected, but this might lead to un-diagnosable faults. In use, none of the cells in the symptom-fault matrix turn red when the flow meter measurements are de-selected, which indicates that no faults are precluded by not using the flow meter measurements, i.e. there is always an alternative symptom available.

The user can request an exhaustive search for the next best n measurements that provide the maximum number of fault detections. The search space can be large so the application firstly will inform the user of the search space size. In the example of figure 8 these are as follows:

1. 21
2. 210 (as selected in the example of figure 8)
3. 1330
4. 5985
5. 20349
6. 54264
7. 116280
8. 352716

Each of the measurement sets of the requested search is listed with any measurement sets that are a superset of the best measurements using fewer measurements can be highlighted, i.e. in a lighter font. This distinguishes measurement sets that can be produced by adding measurements in sequence from a shorter best solution from those where allowing more measurements opens up a different set of measurements (usually for a different aspect or function of the system).

The user is able to select the sets of measurements from the lists shown in figure 8 and can immediately see the affected measurements, symptoms and faults highlighted in (e.g. yellow) on the matrices and the lists. These can then be selected or rejected as required. It can be seen on lower right of the figure that by adding one additional measurement six faults can be detected (i.e. the left pressure sensor detects 6 blockage faults in the left system and the right pressure sensor detects 6 blockage faults in the right system). However, it also possible to detect 80 faults by adding two measurements. Selecting on the Total 6 measurements message expands it to display all measurements involved in any pairs that provide these 80 faults, as shown in figure 9.

The skilled user will appreciate that there are two

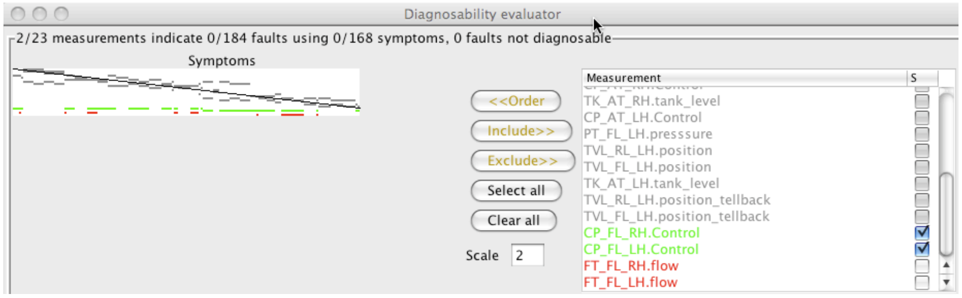

Figure 7: Fuel system - Control valves selected

groups of faults that can be detected (left and right variants). Considering the first set of faults, it is apparent that the flow meter measurement is common, plus either of the left flow or return valves. An engineer would know that both valves are, in fact, mechanically slaved and so the measurements are equivalent, save for a mechanical linkage failure[2]. If it is known that the flow valve is most closely connected to the actuator and return valve slaved to it then this is the one to choose. Thus, the flow left and right meters and flow valves are selected as it is pointless to diagnose only left or right systems. When this is done, it can be seen at the top of the resulting window shown in figure 10 that 116 of the 184 faults are now diagnosable using 6 measurements, and these are shown as diagnosable (green) in the lower matrix and fault list when this is scrolled. Viewing a schematic of the

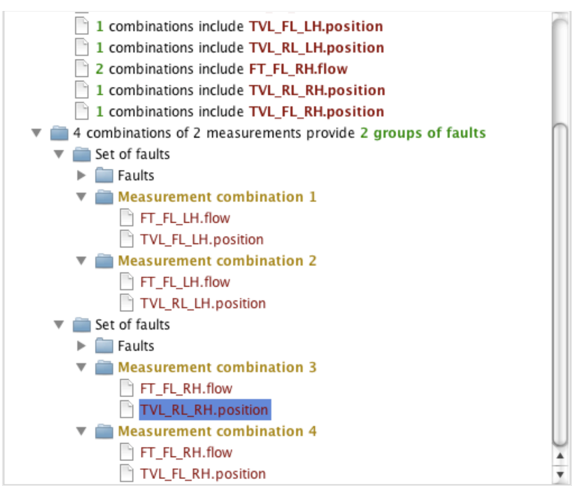

Figure 9: Fuel System - Equally good measurements

system colour coded to indicate diagnosable faults will clearly show that the main fuel and supply return faults are detectable with the subset of symptoms selected at this point. The skilled user/engineer can continue this process of selecting measurements and reviewing the resulting symptom/fault displays until an optimal selection of measurements is made, ideally one that results in all faults being diagnosable with no fault being un-diagnosable using a minimal number of measurements.

It is possible to include features other than simply the number of faults diagnosed in the definition of best measurements, e.g. the ability of the diagnostic system to isolate faults based on the number of different sets and intersections of sets of faults diagnosed by each symptom. Weighting of measurements and/or faults according to physical features such as cost, accessibility or severity is also possible where such data can be obtained, and will result in modified orderings and selections.

6 SYSTEM INSTRUMENTATION

The aircraft fuel system example in the previous sections of this paper had a predefined set of sensors and observable settings. For other systems the task may be to determine which sensors to add to build a diagnostic system. We concentrate on sensors that measure system parameters within the domain of the simulation, so for example in an electrical network rising temperatures as a fault symptom could not be produced as a symptom unless the simulation were to include a thermal model.

It is easy to allow the diagnostic generator to have access to any system (simulation) parameter, and as an example we present an automotive daylight running lights system (DTRL) allowing the current in every wire in the system as a possible sensor input. Perhaps unsurprisingly, many symptoms are generated based on one output observation and a small set of input that are the triggers for the functionality that will cause activity at the observation point. The matrices show which observations are diagnostically equivalent for various sets of faults, for example the vertical 'stripe' patterns in the figure 11 fault - symptom matrix. Figure 11 also demonstrates critical input as a long horisontal bar in the center of the measurement matrix (lighting switch position), without which most faults cannot be diagnosed. The bar is (green) light coloured because it is clear it must be selected for the majority of the symptoms to be usable. The lower right of the figure

[2]the mechanical aspects of the system are not modelled or included in the FMEA in this example

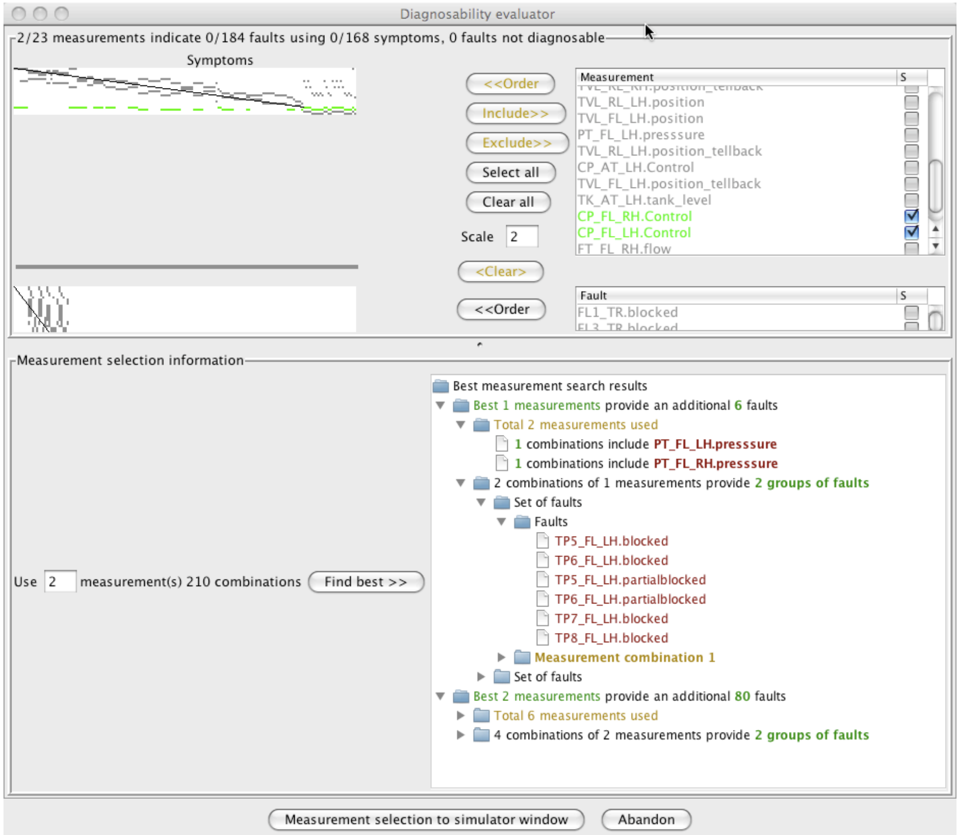

Figure 8: Fuel System - Result of search for two additional measurements

also demonstrates a situation where three equivalent alternative measurements may be used. The number plate lamps have been excluded because they are not directly observable by a sensor, leaving a choice between W16 and W27. W27 was chosen and we note that this makes 3 symptoms redundant (red), although there is no effect on the number of faults that can be diagnosed.

In Figure 12 the remaining elements have been diagonalised on the fault symptom matrix and groups of related faults are clearly seen, each block tends to be related to a different system function, due to structural locality. Hovering the mouse over each block and looking at the symptom conditions easily reveals the states of the system involved, for example the block under the mouse pointer is related to the sidelights and the yellow (light coloured) selected symptoms are all related to the dip lights. Following the process until all faults are accounted for results in the statistics in table 3. Most systems exhibit this law of diminishing returns as more sensors are required to identify fewer faults.

Measurements (55 total)	Faults (46 total)	Symptoms (87 total)
2	17	2
3	19	4
4	28	6
5	35	8
6	38	10
8	42	11
9	43	13
10	44	15
11	45	16
12	46	18

Table 3: DTRL sensor selection

7 CONCLUSION AND FUTURE ENHANCEMENTS

The work presented in this paper builds on the recently developed capability to develop symptom sets based on an automated simulation based FMEA. It provides an engineer with tools to investigate the diagnostic ability of a system or product based on existing or additional sensing. Both on board and workshop diagnostic systems could be produced and evaluated by modifying the visibility of the available observations. The tools have been applied to a number of systems

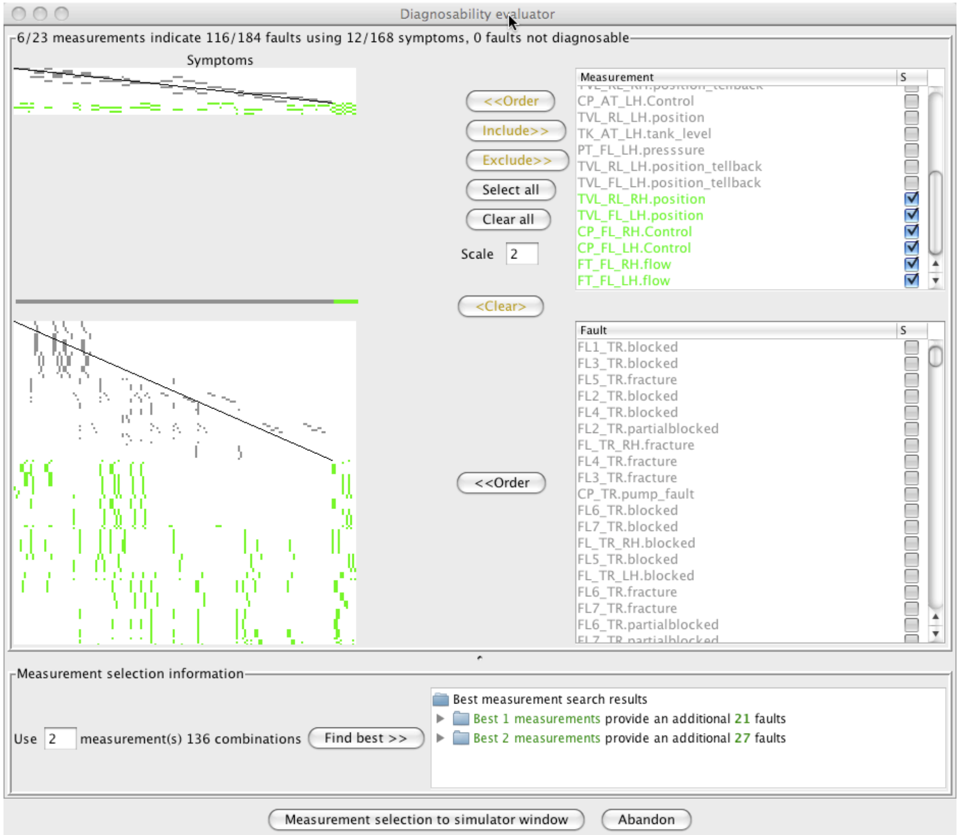

Figure 10: Fuel system - left and right main fuel supply diagnosed

including an aircraft fuel system containing 98 components and 239 possible faults [Snooke07] and a number of automotive electrical systems.

Sometimes diagnostics require specific computations or information from additional domains and these cannot be included unless the system simulation produces the relevant measurements. For specialist diagnostic data it is possible to include a module into the system that produces any such computed results using the usual component modeling capabilities including state machines and general computations. The symptom generator will then utilize any of these specialist measurements that fulfill a diagnostic capability, allowing an engineer to experiment with a number of possible specialist measurements, to determine how well they perform. Some systems contain distinct operating modes and symptoms often relate to specific modes only due to their condition expressions. These modes could be identified and included in the diagnostic generation process to allow choices to be made concerning when faults can be detected during system operation. A good deal of this information is already contained in the functional description of the system and it may therefore be possible to indicate selected information on the matrices via additional colouring or symbolism.

The tool concentrates on optimizing the total number of diagnosable faults. In some applications the ability to isolate faults (to a replaceable unit) and the ability to diagnose faults in specific operating modes is important. In addition there are a number of ranking measures that may be available for fault types, or component failure instances, or affected system functions, all of which could be used to guide the sensor selection advisor. These additions are feasible future additions to the tools that would allow a more tailored diagnostic system to be generated.

There are a few additions to the graphical interface that would improve the tool, for example the ability to select elements by region in the matrices, and to present lists of the elements within these selected regions for inclusion or exclusion.

8 ACKNOWLEDGEMENTS

Aberystwyth Universitys work on the ASTRAEA project is funded by the Welsh Assembly Government, by BAE Systems and by Flight Refuelling Limited. The ASTRAEA project is co-funded by the Technology Strategy Board's Collaborative Research and Development programme, following an open competition. The DRTL system was kindly provided by Sumitomo Electrical Wiring Systems Ltd. This

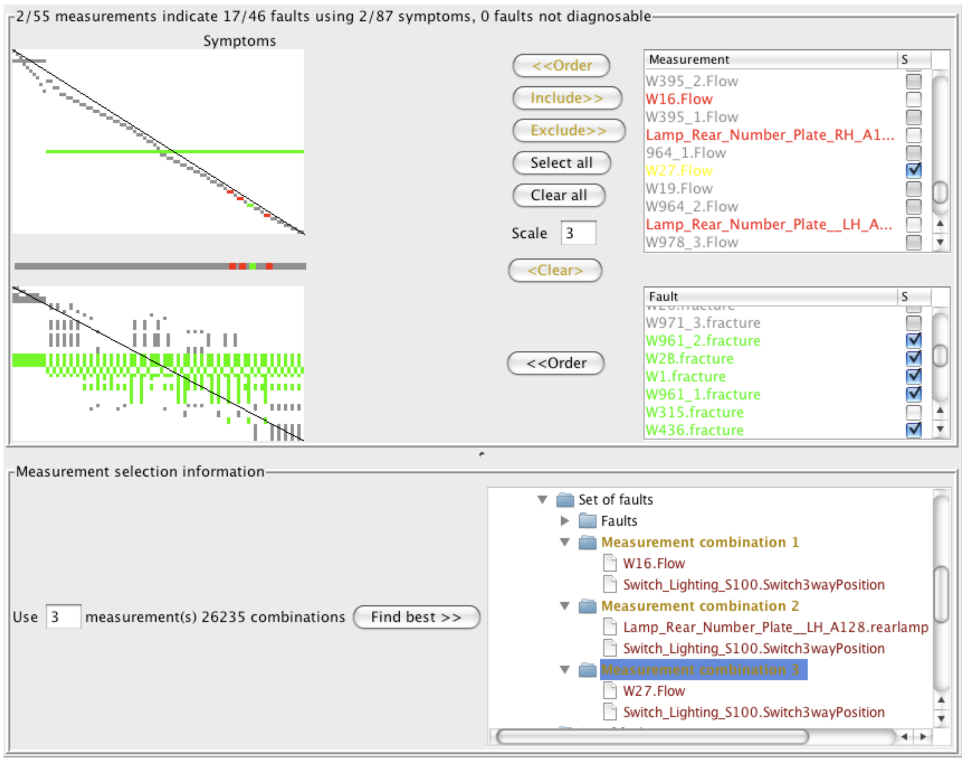

Figure 11: Instrumented DTRL system

work is protected by BAE systems patent applications (0910145.2, ??).

REFERENCES

(ASTRAEA, 2009) ASTRAEA. http://www.projectastraea.co.uk/, April 2009.

(Bell and Snooke, 2004) J. Bell and N. A. Snooke. Describing system functions that depend on intermittent and sequential behavior. In *Proceedings 18th International Workshop on Qualitative Reasoning, QR2004*, pages 51–57, 2004.

(Bell *et al.*, 2007) 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.

(Debouk *et al.*, 1999) Rami Debouk, Stéphane Lafortune, and Demosthenis Teneketzis. On an optimization problem in sensor selection for failure diagnosis. In *in Proc. of the 38th IEEE Conf. on Decision and Control*, pages 4990–4995. University of Michigan, 1999.

(Lee and Ormsby, 1991) Mark H. Lee and Andrew R. T. Ormsby. A qualitative circuit simulator. In *Second Annual Conference on AI Simulation and Planning in High Autonomy Systems*. IEEE, 1991.

(Lee, 2000) Mark H. Lee. Qualitative modelling of linear networks in engineering applications. In *Proceedings 14th European Conference on Artificial Intelligence ECAI 2000*, pages 161–165, Berlin, 2000.

(Maul *et al.*, 2007) William A. Maul, George Kopasakis, Louis M. Santi, Thomas S. Sowers, and Amy Chicatelli. Sensor selection and optimization for health assessment of aerospace systems. Technical Report NASA/TM—2007-214822, NASA, http://gltrs.grc.nasa.gov/, 2007.

(Mushini and Simon, 2005) R. Mushini and Dan Simon. On optimization of sensor selection for aircraft gas turbine engines. In *18th International Conference on Systems Engineering*, pages 9–14. ISBN: 0-7695-2359-5, August 2005.

(Price *et al.*, 1997) C. J. Price, D. R. Pugh, N. A. Snooke, J. E. Hunt, and M. S. Wilson. Combining functional and structural reasoning for safety analysis of electrical designs. *Knowledge Engineering Review*, 12(3):271–287, 1997.

(Price *et al.*, 2003) Christopher J. Price, Neal A. Snooke, and Stuart D. Lewis. Adaptable modeling of electrical systems. In Paulo Salles and Bert Bredeweg, editors, *Proceedings of 17th International Workshop on Qualitative Reasoning (QR2003)*, pages 147–153, Brasilia, Brazil, 2003.

(Price *et al.*, 2006) C. J. Price, N. A. Snooke, and S. D. Lewis. A layered approach to automated electrical safety analysis in automotive environments. *Computers in Industry*, 57(5):451–461, 2006.

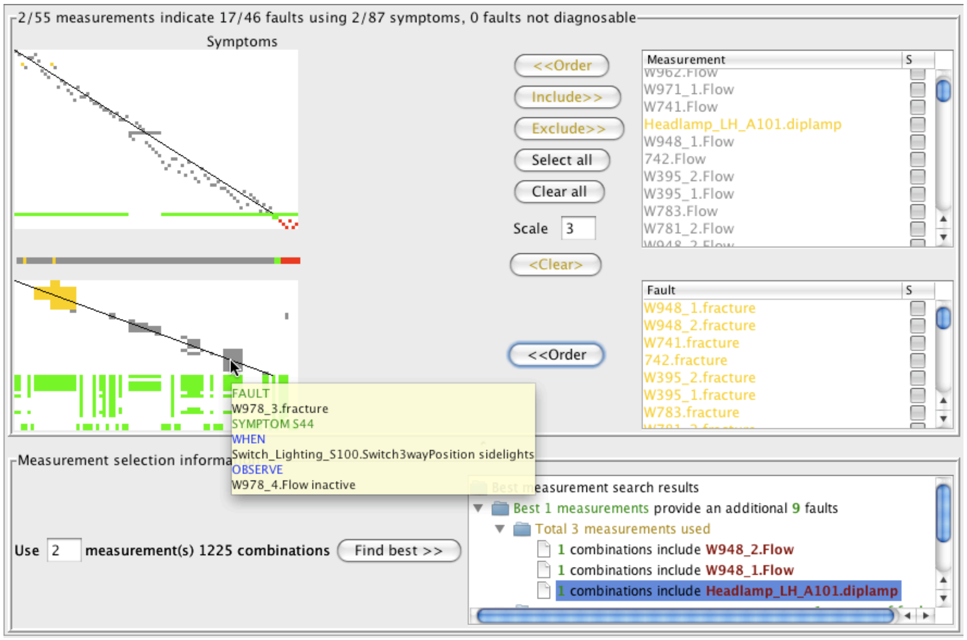

Figure 12: DTRL fault symptom relationships

(Snooke and Bell, 2002) Neal A. Snooke and Jonathan Bell. Abstracting automotive system models from component-based simulation with multi level behaviour. In *Sixteenth International Workshop on Qualitative Reasoning (QR02)*, pages 151–160, Barcelona, Spain, 2002.

(Snooke, 2007) N. A. Snooke. M^{2}cirq: Qualitative fluid flow modelling for aerospace fmea applications. In *Proceedings 21st international workshop on qualitative reasoning*, pages 161–169, 2007.

(Spanache *et al.*, 2004) Stefan Spanache, Teresa Escobet, and Louise Travé-Massuyès. Sensor placement optimisation using genetic algorithms. In *Proceeding DX04*, pages 179–183, 2004.

(Thompson *et al.*, 1999) H. A. Thompson, A. J. Chipperfield, P. J. Flemming, and C. Legge. Distributed aero-engine control systems architecture selection using multi-objective optimisation. *Control Engineering Practice*, 7(5):655–664, 1999.

(Trave-Massuyes *et al.*, 2006) L. Trave-Massuyes, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy relations. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(6):1146–1160, 2006.