An Automated Failure Modes and Effect Analysis Based Visual Matrix Approach to Sensor Selection and Diagnosability Assessment

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 (FMEA) using qualitative model based reasoning techniques. The automated FMEA provides a comprehensive set of fault–effect relations by qualitative simulation and can be performed early in the design process. The comprehensive nature of the automated FMEA results in a fault-effect mapping that can be used to investigate the diagnosability of the system. A common requirement is to facilitate cost reductions by removing sensors or to improve diagnosability by including additional sensors. Measurements are typically expensive (in the broadest sense) and the problem addressed by this paper is how to allow select a set that fulfills the diagnosability requirements of the system. This paper documents a technique that provides an engineer with easy access to information about diagnostic capability via a matrix visualisation technique. The focus of the work was for 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.

Automated failure mode and effects analysis (FMEA) is a technique that is used to provide a comprehensive and consistent description of the effects of component faults (Price *et al.*, 1997; 2006). The results can be used to generate symptoms for an onboard diagnostic application. Conceptually this is the process of generating an effect → fault mapping from the fault → effect mapping provided by the FMEA while excluding effects present within nominal observations. It is not the purpose of this paper to describe in detail the transformation of the FMEA into a set of diagnostic symptoms but rather to use the symptoms to assist diagnosability assessment. The FMEA based diagnostic system has a comprehensive fixed set of symptoms that detect as many of the faults itemised in the FMEA as possible, and is more closely related to a manually coded set of diagnostics than traditional run time consistency or abductive MBR approaches (Peischl and Wotawa, 2003) that compare the results of running an on-board system model with actual observations (Struss and Dressler, 2003; Struss, 1992; Reiter, 1987).

The proposed diagnostic system is limited to the operating modes considered in the FMEA, and it can only detect faults defined in the component library but it does have advantages of fast on-board execution, comprehensive analysis of all possible symptoms, and behaviour that can be validated for certification purposes. This allows combinations of measurements to form symptoms that may not be immediately obvious to an engineer and in any case would be very tedious (and potentially error prone) to generate manually. Symptoms are generated effortlessly since no additional modelling is required beyond that to produce the FMEA. The FMEA requires a library of components with failure modes, system schematic, and operating scenaro (Price *et al.*, 1997). The system behaviour is simulated from a schematic based structural model and compositional component (nominal and failure) behaviour models. This ensures the correct (qualitative) effects are available for structural and functional failures thus allowing wider range of failures than a purely structural model, but without the arbitrary modelling decisions associated with manually produced causal models (Console *et al.*, 1989). While multisignal and dependency modelling approaches such as TEAMS-RT (Deb *et al.*, 1995) allow sophisticated test

sequencing and do not require fault models thus allowing unforseen faults to be diagnosed, there is substantial modelling effort required specifically to support the diagnosis and diagnosability investigations in this approach.

It is useful to note that the qualitative simulation means that exact systems parameters are not required allowing a broad analysis early in the design cycle. For example we may have the topology of a new aircraft fuel system design concept but the pipe lengths may not yet be known, however the symptom *when valve a open and pump b on then pressure in pipe x low* for potential faults {*blockage in pipe y*, *valve z stuck open*} can be generated. In the Automotive industry electrical systems suffer from these issues to an even greater extent, where system and harness designs must be proposed and analysed for failure characteristics prior to the availability of detailed spatial or component parameter information. Qualitative measurements allow broad regions of system (mis)behaviour to be treated by a relatively small set of symptoms which is good for assessing broad diagnosability issues. Using the qualitative rules on-board requires some additional work and in practice measurement thresholding and a Bayesian network that allows weighting of fault types and measurement reliability has been used to rank diagnoses for the ASTRAEA project, however for the purpose of investigating diagnosability at design time for a topologically complex electrical or fluid flow system, the qualitative regions of behaviour provide relevant details based only on logical diagnostic espressions of qualitative measurements.

An onboard diagnostic system will only have access to a limited number of measurements, and the ability to rapidly investigate at early design stages which measurements may be useful for fault detection is valuable. When many hundreds of these symptoms are possible, each requiring selections of measurements, we find that by providing or excluding measurements the set of usable diagnostic rules and hence system diagnosability and isolatability is changed. Measurements are typically expensive (in the broadest sense) and the problem addressed by this paper is how to allow select a set that fulfills the diagnosability requirements of the system.

The Automated FMEA report itself contains a high level description of fault effects in terms of the failure of system function, however more detailed information concerning every variable and signal in the system is produced by the simulation, and diagnostic rules can therefore be generated utilising very detailed information. In most systems there are various costs (financial, mass, layout, harness complexity) involved with each sensor, resulting in a need to compromise between diagnostic ability and sensing and therefore a small set of the most useful and obtainable measurements need to be selected. 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. Typical issues that require consideration 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.

Some existing optimisation methods are very specific solutions to an individual system e.g. (Maul *et al.*, 2007; Mushini and Simon, 2005) and do not support schematic and component library based analysis. Other approaches are generic but require large modelling effort to enable varied additional application specific information to be taken into account (Debouk *et al.*, 1999; Trave-Massuyes *et al.*, 2006). Even when the information required to assess diagnosability can be modelled, 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). Experience shows that in many cases there are simply too many additional application specific considerations that an engineer can resolve but which would be difficult to provide to a fully automated system. For example spatial constraints associated with adding new sensors for electrical systems where an engineer may have a good idea where it is feasible to add sensors, but without a detailed and 3D spatial model integrated with the electrical circuit description it is impossible for an automated system to decide. A second example is the knowledge of which sensors are required for basic system functionality and therefore have a very low cost to any diagnostic system and those which are present for diagnostic purposes only. A system engineer will know this due to his in depth functional and causal understanding of the system architecture but it is very difficult to extract this information from an electrical circuit diagram. As a final example an engineer may know that some parameters are very noisy and should perhaps be avoided (or require additional processing) as inputs to a diagnostic system for example a fuel level sensor on an aerobatic aircraft. Modelling could be provided for all of the above situations however the investment in modelling is high for relatively low return, and we take the alternative approach of providing tools that support relatively simple models but allow the engineer to easily make decisions and understand the effects on the potential diagnosability of the system.

The following sections of this paper firstly outline the FMEA generated 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 matrix approach is presented to assist an engineer to quickly visualize the diagnostic behavior of the system. This allows rapid investigation of the sensor selection and placement options available. The technique has been used on several case

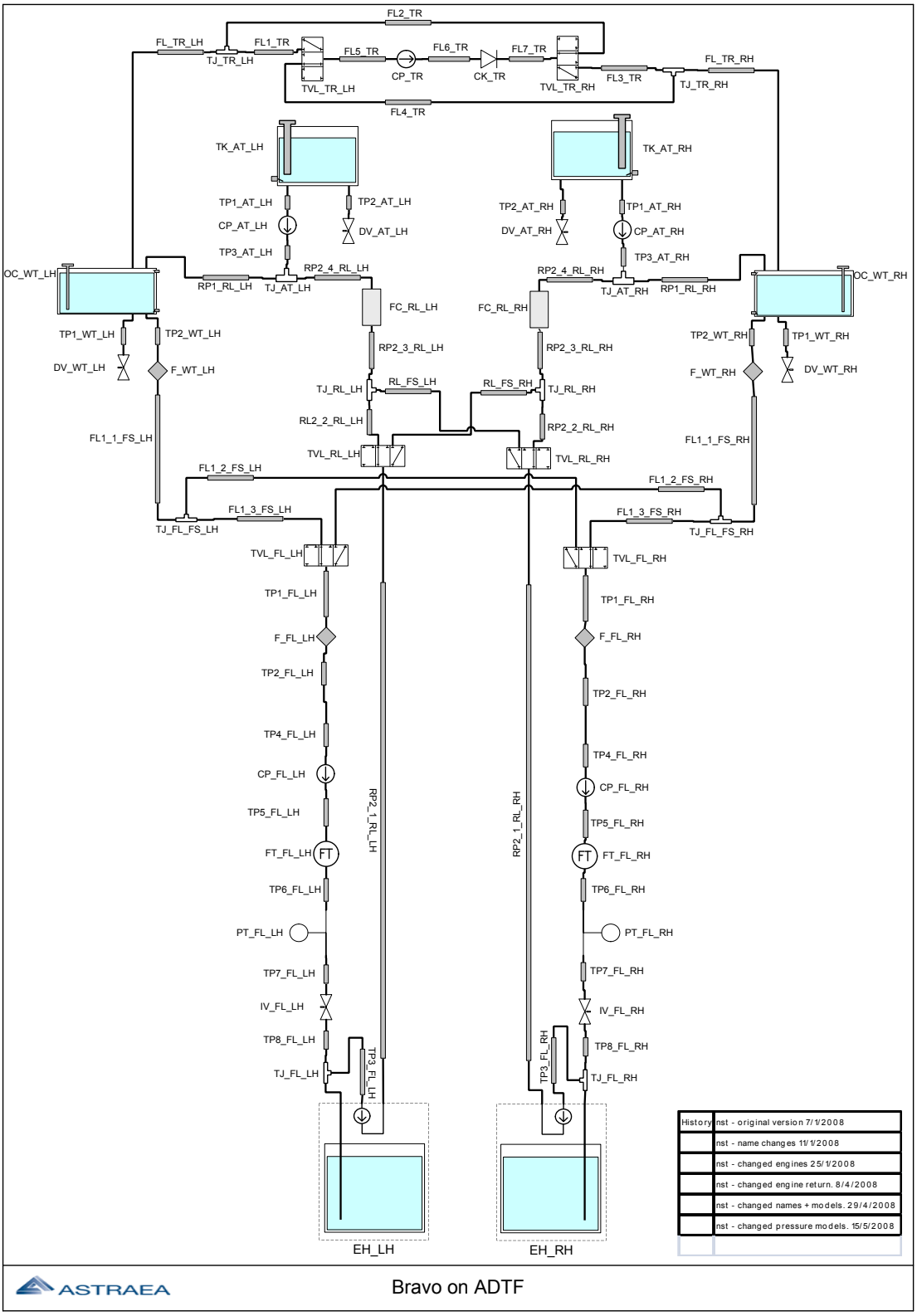

Figure 1: Fuel system schematic

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.

Table 1: Symptom states

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

The third row illustrates a 'negatable' symptom able to exonerate faults ($\neg F$) and is the reason for Ce expressions. We have observed that allowing negatable symptoms typically leads to fewer symptoms but requires 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.

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 create 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 symptom examples for the system in Figure 1 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.

Based on the systems analysed for automated FMEA from the automotive application areas we find there are typically hundreds of qualitatively distinct faults (several for each component) and several potential measurements associated with each component (Price, 2000). Although a symptom can predict any number of faults, and a fault can be predicted by any number of symptoms, we have found that the number of qualitative symptoms generated is of the same order as the number of faults considered in the FMEA. Informally this seems to be for the following reason. Useful symptoms do not require more than few measurements, and in fact symptoms that require many measurements ($>$10) are disallowed because they generally occur due to artifactual issues associated with incomplete exercising of the system state space by the FMEA, or due to approximations in the component behaviour modelling. In addition if a reasonable level fault isolation is possible (and we assume it is given the above observation of several measurements per component), symptoms on average predict a relatively small number of faults and because symptoms are generated to be as specific as possible each fault is on average predicted by a relatively small number of symptoms. Therefore on average the number of symptoms is of the same order as the number of faults, and since symptoms may require several measurements but measurements are on average present in more than one symptom the number of measurements is also of the same order as the number of faults. For these systems it is feasible to use visual matrices depicting measurement-symptom-fault relationships as proposed in the next section.

An example automatically generated diagnostic system with 168 symptoms produced from an automated FMEA is illustrated in Figure 2 for a twin engine aircraft fuel system in Figure 1 with 184 possible faults. 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 and 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 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 $Ce = \mathrm{true}$. The symp-

Table 2: Example symptoms

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

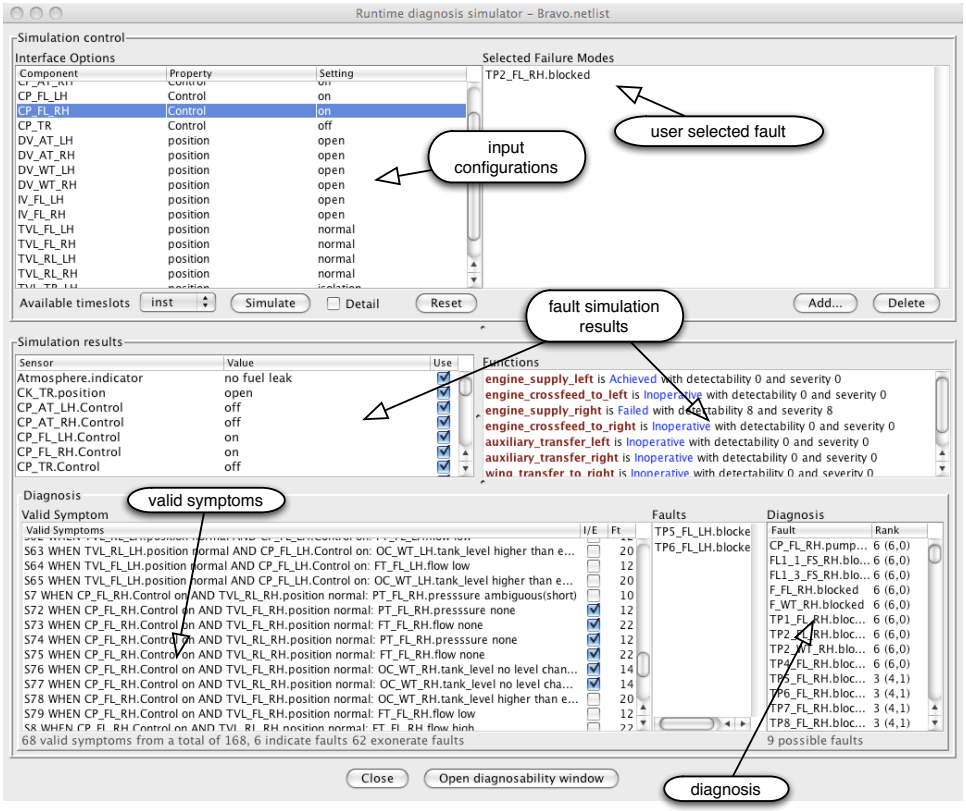

Figure 2: Diagnostic evaluator interface

tom 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 nine 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 exercised. 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 enable an engineer to decide on a set of sensors which will cover all possible faults on the system, due to the number of 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 by a generic example in Figure 3. This Figure is intended only to show the form of the matrices, for a real system there may be hundreds of rows and columns, and it is

the visual correlations present in the matrices that provide information to the engineer (zoom/pan is available for larger matrices). A graphical method to assess competing requirements was also described by (Thompson *et al.*, 1999) however this was aimed at architectural choices rather than sensor selection.

Each symptom is represented by a column in the matrices on the left of the Figure. In the upper matrix any measurement included in the Ce or Oe expression for a symptom is indicated by non empty element in the column representing that symptom. In the lower matrix the set of faults indicated by each symptom is indicated by a non empty element in the column representing that symptom. The top matrix shows which

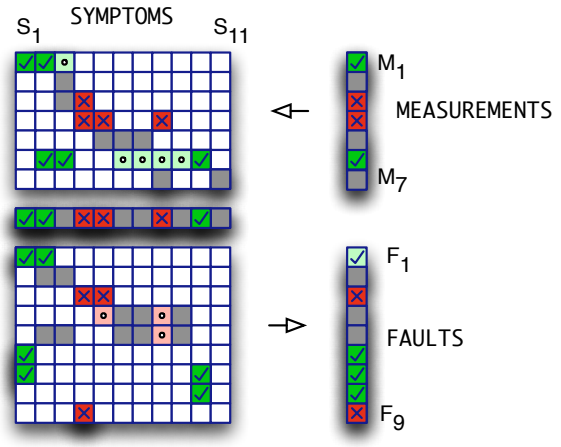

Figure 3: Measurement - Fault Matrix

measurements are required for each symptom and the lower matrix shows the faults that each symptom can diagnose. A colour coding system is used to indicate the status of each element and these change as additional measurements are included or excluded in the measurement vector (top right). Green indicates that an item is available to the diagnostic system (also a small tick is shown for clarity) and grey indicates that the item forms part of a diagnostic relationship but is not available because it needs a measurement not yet observable to the diagnostic system. Red is used to explicitly exclude items - for example when the engineer has decided that a measurement is not available.

Once a measurement is made available it appears as a green element in the measurement vector (top right) and also as green elements for the symptoms that require it in the corresponding row in the top matrix. Any symptoms that have all the necessary measurements available to evaluate their Ce and Oe expressions have the appropriate element coloured green in the row vector on the (centre left) indicating that the symptom can be fully evaluated and hence used to diagnose its associated faults. The lower left matrix column elements represent associated faults diagnosable by an available symptom, and are coloured green to indicate a diagnosable fault. Finally any faults that have one more more available diagnosing symptoms are coloured green in the faults vector (lower right). A lighter green colour (centre dot) in the top matrix indicates that a measurement is available to a symptom but the symptom requires further measurements.

If a measurement is excluded by the engineer 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. Hence, cells that are pink (dot) in the lower matrix indicate a symptom that cannot be used for a fault that can be diagnosed using an alternative. Elements that form diagnostic relationships but are undecided are coloured grey and may therefore be included or excluded based on the availability of undecided measurements. These 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. The GUI follows a similar structure to Figure 3 with the measurement-symptom matrix at the top left and the symptom-fault matrix lower left. The measurements and faults are now shown as textual lists with the order of the lists being the same as the rows in the matrices. Measurements can be selected or deselected using the lists and the associated fault status is updated, together with the colour coding of the matrices. The yellow colour is used to allow sets of measurements to be proposed prior to committing or excluding them, allowing the incremental change in faults that can be diagnosed to be observed.

The visible patterns in the matrices are formed by the structure that exists in the fault behaviour of the system. The patterns represent correlations between measurements, faults and symptoms. In addition the matrices are relatively sparse as expected since measurements, symptoms and faults form (overlapping) sets and subsets due to the structure of the system and the predictable behaviour of the system in the presence of faults. Specific 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. In practice we find measurements (inputs) such as valve positions and switches that affect major system state typically have this characteristic.

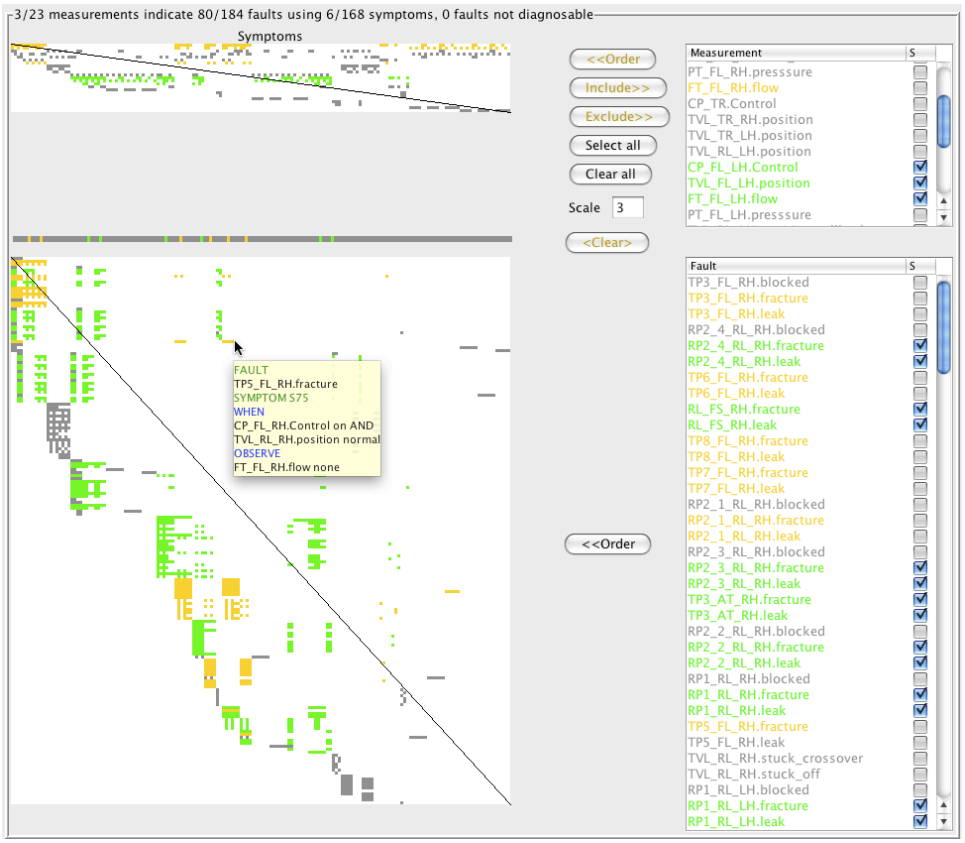

Figure 4: Aircraft fuel system matrix example

- Highly populated columns in the measurement-symptom matrix indicate symptoms that require many measurements.
- Highly populated columns in the fault - symptom matrix indicate symptoms that can diagnose many faults. These are the symptoms that provide cheap detectability, but poor fault isolation.
- Similar patterns in several fault - symptom columns show that there may be a choice of symptoms that diagnose the same set of faults.

The ordering of the measurements, symptoms, and faults will change the appearance of the matrices and where possible we would like to group related symptoms and faults into rectangular blocks that represent alternative symptoms that have equivalent diagnostic power. Reordering the matrices is discussed in section 4.1.

4 SENSOR SELECTION

Simply by selecting and deselecting measurements at any point in the measurement selection process an engineer can find out which (additional) measurements provide the ability to detect many faults in the context of the currently available measurements. In Figure 4 the user has already selected some measurements using the tick boxes and the result of this in terms of the symptoms and faults that can be diagnosed is shown as green elements (darker) and as tick boxes in the fault list.

There are usually a set of measurements that will definitely be available to the diagnostic system, and some that the engineer knows will be important in the diagnosis of a required set of faults and these can be selected. At some point the question will arise as to the next set of measurements that diagnose the maximum number of faults.

The problem of finding n additional measurements that allow the maximum number of faults to be detected is exponential in the number of additional measurements if a brute force search is carried out. Due to the localisation of measurement - fault relationships it is only useful to use small numbers for n, until a new 'block' of elements (measurements, symptoms and faults) is identified. For an exhaustive search if n is the number of additional measurements required and r is the number of unselected measurements remaining there are $\frac{r!}{(n*(r!-n))}$ combinations of measurements to consider. We has observed that in the early stages systems tend to have a few critical measurements that provide big diagnostic returns and so a relatively small n is adequate to find these, and once a good number of the measurements are determined, r becomes small al-

lowing larger n in reasonable time, although by this stage symptoms and faults tend to be closely coupled, so adding a measurement gains a few additional faults, and therefore the next best n measurements provides a superset of the faults that can be obtained by the next best $n-1$ measurements.

The best solution may not necessarily be included in best solutions for larger numbers of measurements so strict hill climbing solutions do not work in general and allowing the engineer to choose n based on the visible structure of the matrices provides a reasonable compromise. No attempt has been made to improve the search using other methods (e.g. backtracking heuristics) because the major issue is that there are often many possible solutions for 'next best' combinations of n measurements often due to symmetry in designs, or sensors equivalent for some diagnostic aspect, or simply different parts of the system structure all of which require the same number of (different) measurements. 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 so a symmetrical left and right of sensors would be added eliminating the alternative left or right permutations from the next set of measurements. Our experience is that it is better to only consider a small number of measurements and then investigate why there are alternatives, make a selection (noting any significant effects on the matrices) and then consider subsequent measurements associated with the next region of system structure and behaviour.

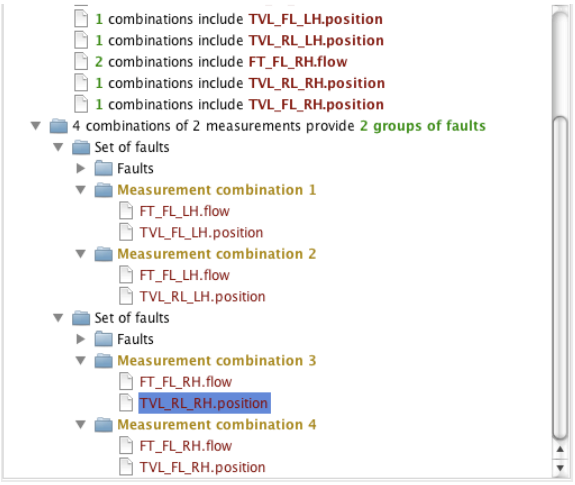

Figure 5: Fuel System - Equally good measurements

The results of a search for the maximum number of faults diagnosable from n measurements naturally fall into a hierarchy presented to the user that gives access to the alternative solutions.

1. Top elements of the hierarchy specify the maximum number of additional faults can be diagnosed for each additional measurement set size from $1..n$. In the subsequent example in Figure 9, the phrase "Best 2 measurements provide 80 additional faults" is seen in the lower right.
2. For each of the elements in item 1 the total number of *different* measurements involved in any of the possible solutions are listed. For example "Total 6 measurements used" indicating that there are 6 distinct measurements that are used in some combinations (in pairs for best 2 measurements) to form the best solutions.
3. There are often several different sets of measurements that can diagnose exactly the same set of faults. This forms the next grouping under item 1 for example "4 combinations of 2 measurements provide 2 groups of faults" in Figure 5. This means that there are 2 distinct sets of faults diagnosable but 4 different pairs of measurements that have been found that are relevant to the 2 sets of faults.
4. The sets of faults are itemised together with the measurements required for each fault set is given under item 3 showing which different measurements can be used to detect the set of faults. Often there will be several similar sets of measurements with only one different measurement alternative.

The engineer can select any set of measurements at any level in the above categorisation simply by selecting any item in the hierarchy as illustrated in Figure 5. The impact on the symptom set and fault set is shown highlighted in yellow on the matrices as in Figure 4 where a whole set of measurements has been selected. By selecting alternately different groups of measurements the diagnostic effect can be visualised. For example some sets of measurements provide very small changes to a set of faults whereas others may provide for diagnosis of a completely different set of faults. Hovering over the matrices instantly produces a tooltip that identifies what the element represents (see Figure 4).

4.1 The diagonal matrix

To gain a much better understanding of the relationships contained within either matrix they can be automatically reformed into an 'approximate diagonal form' which places all the non empty matrix elements as close to an imaginary line from top-left to bottom-right as possible (this is the purpose of the "Order" buttons on the tool interface). The algorithm used is similar to the well known bubble sort applied alternately to row and columns, with the ordering comparison based on the imbalance of the number of non zero cells from the diagonal. 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 to either side of the imaginary diagonal line across the matrix. Figure 6 shows an example 6 by 4 matrix.

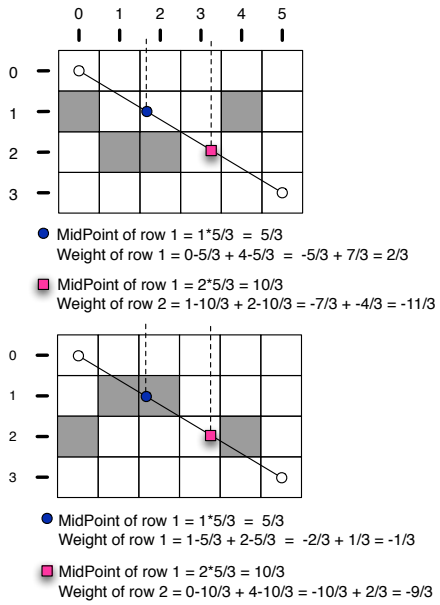

Figure 6: Producing the diagonal matrix

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 6) 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 6, 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 a bubble sort, 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 by producing visual blocks of colour representing the relationship between groups of measurements, symptoms and faults. Disjoint blocks also graphically illustrate parts of the system that are diagnostically separate, for example sets of symptoms and measurements that are the only possibility for diagnosing a set of faults for some part of a system.

Each row or column sort is effectively a bubble sort with a worst and average $O(n^2)$ complexity where n is the number of measurements, or symptoms, or faults dependent of which dimension is being sorted. However the matrices have two characteristics that in practice seem to make the average complexity of the whole algorithm not much worse than this. Firstly the matrices are rather sparse and secondly there is a strong relationship between groups of elements on each axis. For example we find (and expect also) a set of faults that can be diagnosed by a set of symptoms using a set of measurements. The algorithm will only need a single sort on one dimension for a matrix that has a perfect simple diagonal form since the order of one axis can be arbitrary and the elements moved onto the diagonal by reordering the other. The more 'imperfect' the final diagonal matrix in the sense of the number of empty elements between the diagonal and any non zero element in the result, the more iterations of the row and column sort sequence could be needed. This is because the solution may require (worst case) a specific ordering of each axis. The matrices are relatively sparse for the reasons outlined in section 2 and this combined with the systematic effects of faults and the structure of the system cause the matrices to have a good 'compact' diagonal form, and in fact they will only be useful if this is the case. Therefore only small number iterations of the sorting should be required this has been observed experimentally. We also observe that the algorithm is converging towards the solution and therefore once the first sort is completed on each axis, subsequent sorts start with most of the elements already in the correct order. The visual effect is that non empty elements 'bubble' along the diagonal until each group of elements has achieved its best order on the diagonal.

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 and do not participate in the sorting. This is why the diagonal line does not extend the full size of the center left matrix in Figure 12 (discussed later) which is also an example of a diagonal symptom-fault matrix showing blocks of elements that represent distinct sets of symptoms that diagnose distinct sets of faults for an automotive sys-

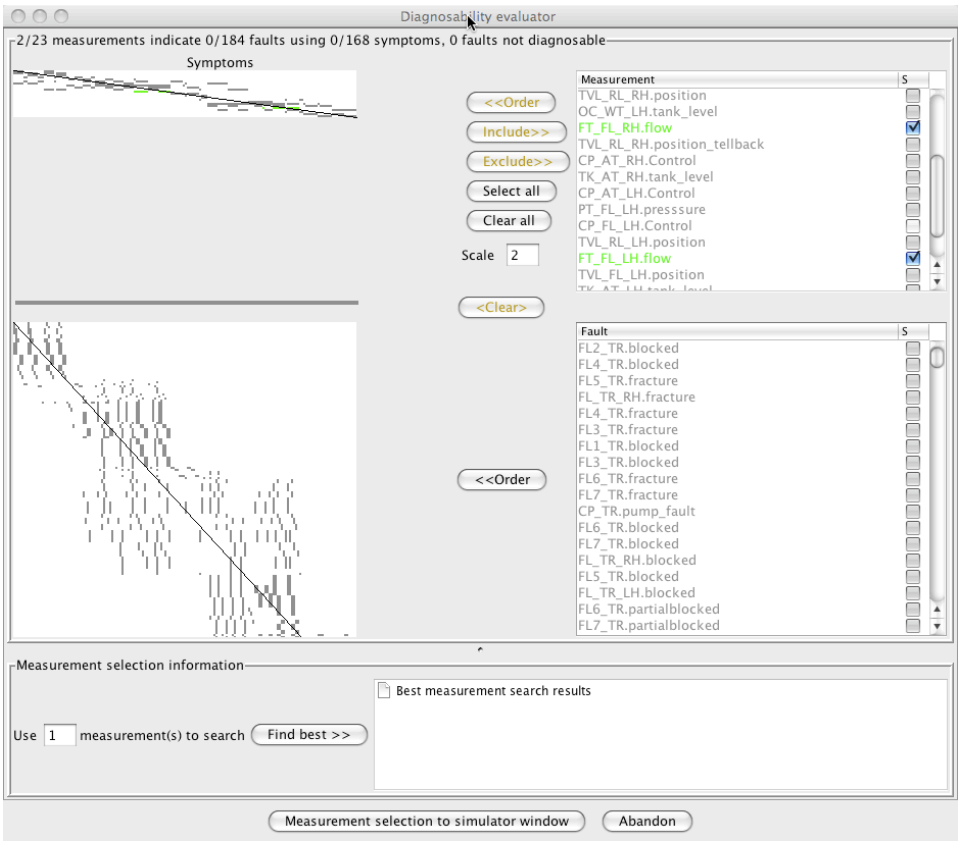

Figure 7: Fuel system - Selected flow measurements

tem. It is useful to repeatedly make the matrices diagonal as an interactive activity during the measurement selection process as diagnostic characteristics are discovered.

5 EXAMPLE

The benefits of the diagnosability matrices 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. In Figure 7 the measurement matrix has been diagonalised and most measuremets set undecided, and we see that the majority of measurements are needed in several symptoms because of the horizontal bars in the matrix. If the user/engineer knows that the measurements from the flow meters are definitely available to the diagnostic system, then these 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 making these measurements available to the diagnostic system would not be enough to allow it to diagnose any fault. The summary at the top of the window notes that we have chosen to make 2 measurements visible but this would not allow diagnosis of any faults (0/184). The information "0 faults are not diagnosable" refers to the as yet undecided measurements and hence by adding additional measurements we could still be able to diagnose all the faults. If measurements are excluded then the number of undiagnosable faults may rise, and some systems may have undiagnosable faults even with all available measurements if the FMEA had faults that provide no observable abnormal effect.

The pump control values are computer controlled and hence available to a potential diagnostic system (the engineer knows this even though the FMEA was only performed on the fluid system), and can be selected, in Figure 8. It can then be seen that these 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 deselected, 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

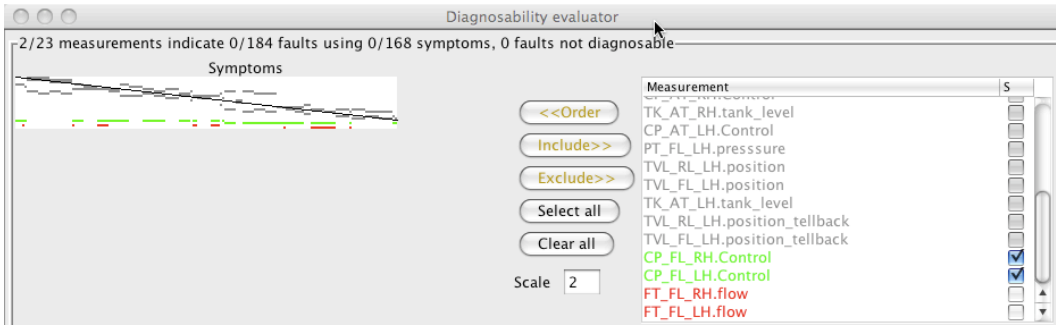

Figure 8: Fuel system - Control valves selected

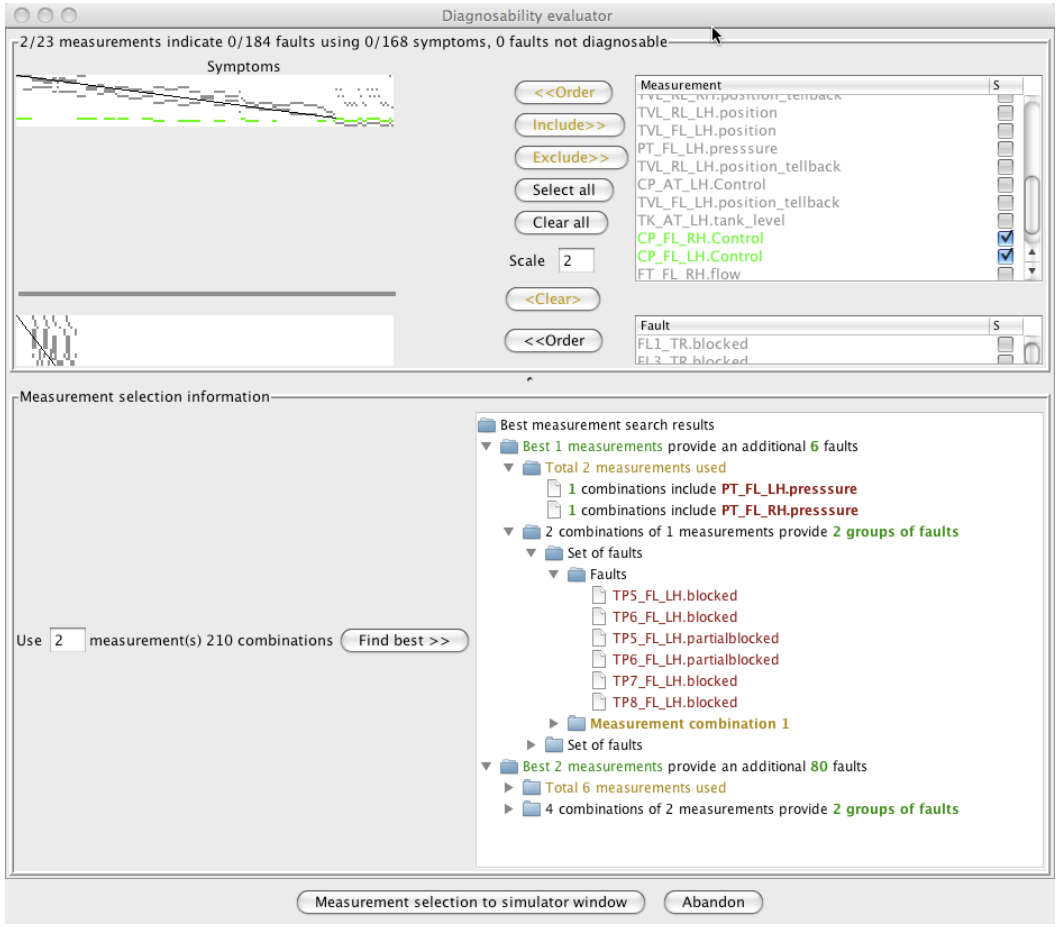

Figure 9: Fuel System - Result of search for two additional measurements

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 9 these are as follows:

1. 21
2. 210 (as selected in the example of Figure 9)
3. 1330
4. 5985
5. 20349
6. 54264
7. 116280
8. 352716

The result of a search for the next best two measurements is seen in Figure 9. The user is able to select the sets of measurements by clicking on any of the items under "Best measurements search results" and will immediately see the affected measurements, symptoms and faults highlighted (not shown in the Figure). At each level all of the measurements related to lower level categories will be selected. Also in this list a darker font is used to distinguish parts of a measurement set that are not part of a shorter best solution. 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 5.

The skilled user will appreciate that there are two 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[1]. 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 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. For systems that include diagnosis specific sensors (e.g. vibration sensors) from other domains, hand crafted or externally generated symptoms can easily be added to the symptom set and included in the overall diagnosability analysis, if required.

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 the function output observations (lamps) and the inputs 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 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 this makes 6 symptoms redundant (red), although there is no effect on the number of faults that can be diagnosed.

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

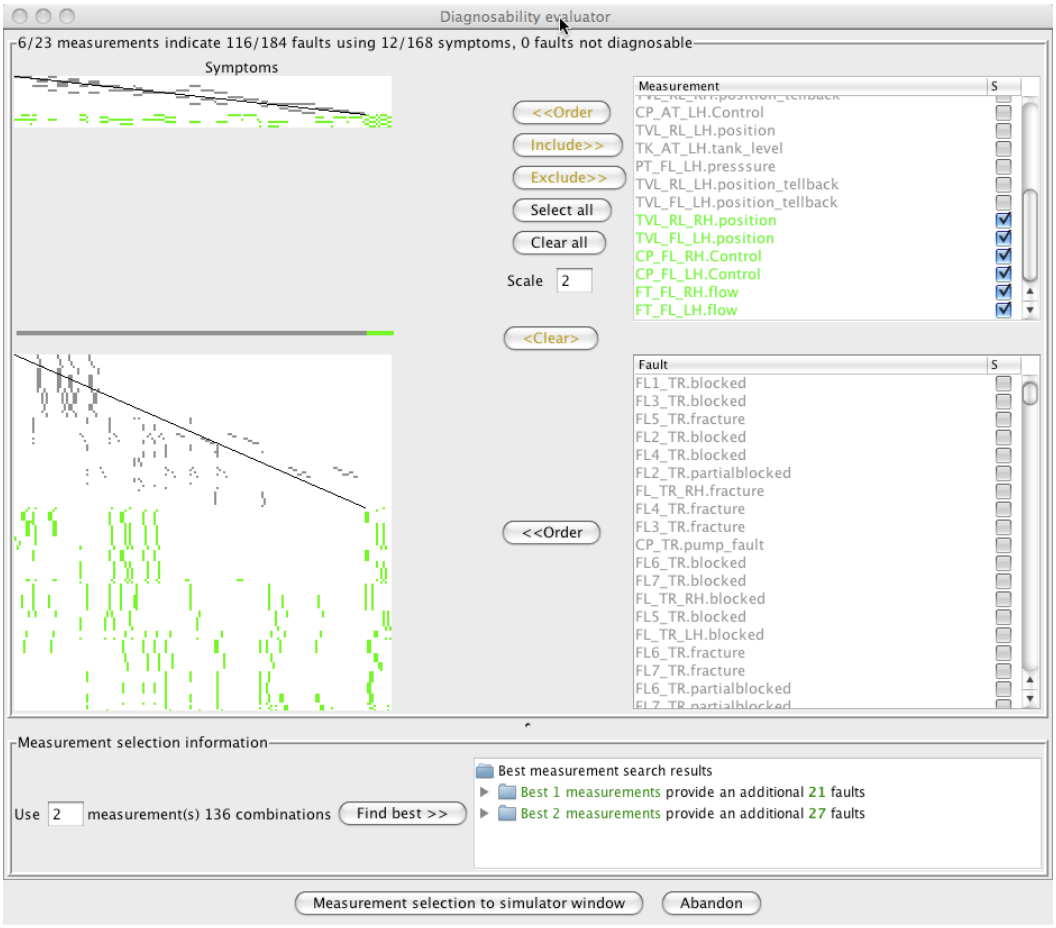

Figure 10: Fuel system - left and right main fuel supply diagnosable

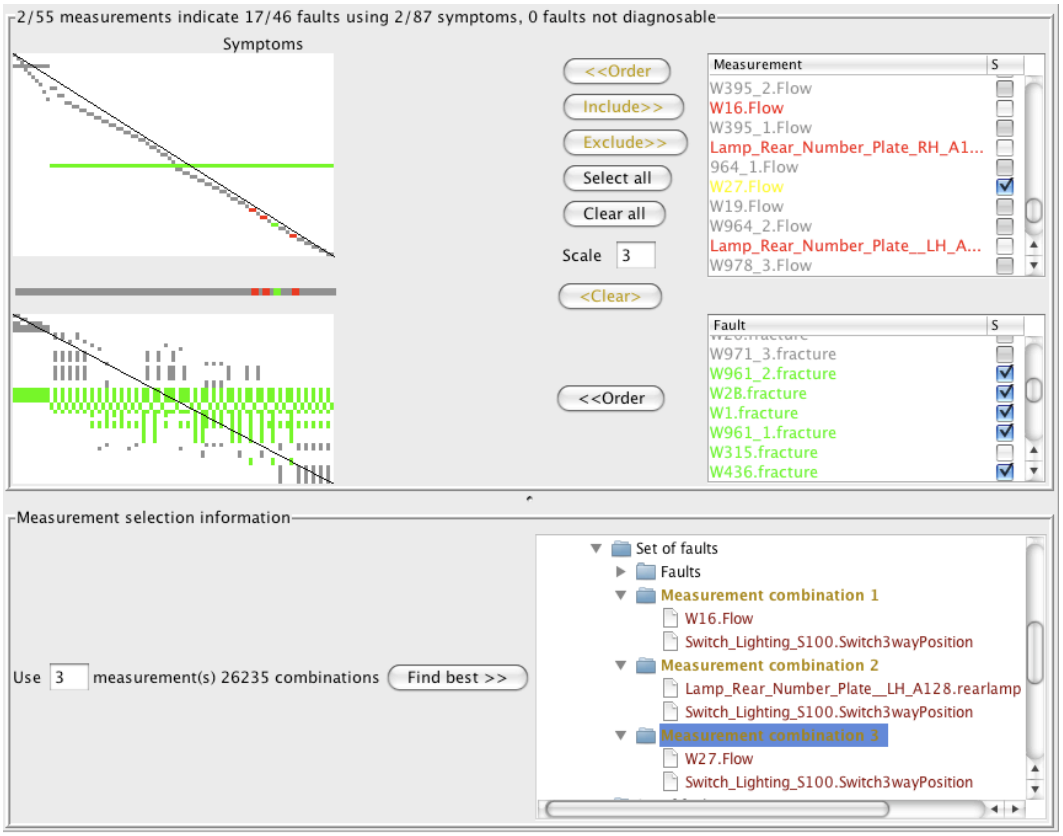

Figure 11: Instrumented DTRL system

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.

Table 3: DTRL sensor selection

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

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 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 spe-

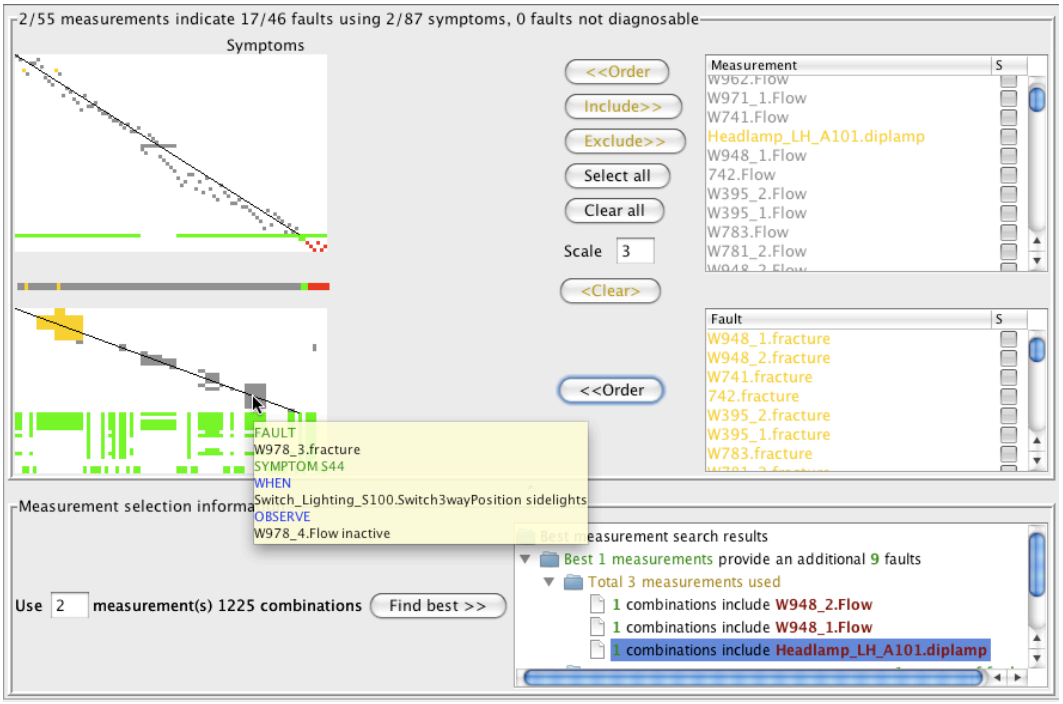

Figure 12: DTRL fault symptom relationships

cialist 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. Various graphical notations could be developed to visualise these relationships by colour or spatial grouping or possibly a hierarchical version of the matrices that allow rows or columns to be aggregated, such an approach may also help in the presentation of very large systems if there are disjoint sections to the diagnostic structures present in the matrices. In addition there are a number of ranking measures that may be available for fault types, 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. The ability to view the current set of diagnosable faults and measurements needed by (for example) colouring or labelling components on the original system schematic as each selection is made may be a useful way of assessing diagnosability.

8 ACKNOWLEDGEMENTS

Aberystwyth University's 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 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.

(Console *et al.*, 1989) Luca Console, Daniele Theseider Dupre, and Pietro Torasso. A theory of diagnosis for incomplete causal models. In *In Proc. 11th IJCAI*, pages 1311–1317, 1989.

(Deb *et al.*, 1995) S. Deb, K.R. Pattipati, V. Raghavan, M. Shakeri, and R. Shrestha. Multi-signal flow graphs: a novel approach for system testabilityanalysis and fault diagnosis. *Aerospace and Electronic Systems Magazine, IEEE*, 10(5):14–25, 1995.

(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.

(Peischl and Wotawa, 2003) Bernhard Peischl and Franz Wotawa. Model-based diagnosis or reasoning from first principles. *IEEE Intelligent Systems*, pages 32–37, 2003.

(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.

(Price, 2000) C. J. Price. AutoSteve: automated electrical design analysis. In *Proceedings ECAI-2000*, pages 721–725, 2000.

(Reiter, 1987) R Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

(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^2cirq: 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.

(Struss and Dressler, 2003) P. Struss and Oskar Dressler. A toolbox integrating model-based diagnosability analysis and automated generation of diagnostics. In *Proceedings of the 14th International Workshop on Principles of Diagnosis*, 2003.

(Struss, 1992) Peter Struss. What's in SD? towards a theory of modelling for diagnosis. In Wlater Hamscher, Luca Console, and Johan de Kleer, editors, *Readings in model-based diagnosis*, pages 419–449. Morgan Kaufman, 1992.

(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.