IDENTIFYING FAILURE MODES RETROSPECTIVELY USING RCM DATA

J. Sikorska (Imes Australia Pty Ltd),
L. Hammond (DSTO),
P. Kelly (CASWA Pty Ltd)

SUMMARY: Managers are constantly being told that the quality of data is insufficient for producing reliable results from business models. Reliability analysis is no exception. In this paper, the authors show that it is possible to retrospectively improve the quality of failure histories stored in computerised maintenance management systems (CMMS). A tool was developed to help reliability professionals assign the most appropriate failure mode to a particular work order. Case-based reasoning is used to contextualise the data and reduce the number of possible options to a list of most likely candidates; the case-bank is developed from FMEA based studies such as RCM. Traditionally, CMMS and FMEA systems have rarely been integrated and differences in emphasis, underlying coding structures and information supplied often make this difficult. However, using diesel engine data from HMAS Anzac, it is shown that these issues can be overcome in a sustainable manner. It is also possible to adapt the system for use in real-time failure troubleshooting.

KEYWORDS: FMEA, RCM, CMMS, failure modes, reliability analysis, data quality, data cleansing.

1. INTRODUCTION

It is widely accepted that data is fundamental to robust and reliable decision-making. In a previous paper [1] the authors developed an eight step process for improving data quality. This process is based on the basic definition of data quality as: '*Data that is fit for purpose*'. The paper suggested that a DQ process should: (1) define the business need; (2) identify metrics; (3) determine the data required; (4) analyse quality (*data profiling/auditing*); (5) identify ways to improve quality; (6) implement changes (*data cleansing*); (7) assess changes; and finally (8) establish review periods. All but steps 4, 5 and 6 should be achievable using an organisation's standard business processes. Due to their very specific application, advanced systems and processes for *data profiling* and *data cleansing* however, are not commonly available.

Wherever possible, the best and easiest option to rectify poor data quality is to monitor, report and prohibit discrepancies as the data is being entered, thus allowing or compelling data collectors to rectify the erroneous information immediately (i.e. before they forget and/or the data is no longer available). There is no doubt that this should be a mandatory part of any data capture process. There is also merit in developing mechanisms to assess how well software manages the data entry process. However, it is unrealistic to expect that any datasets will ever be completely error free, no matter how well intentioned the data collectors and well managed the associated systems, processes and databases [2]. This approach also disregards the wealth of information that presently resides in existing data stores. Therefore, when data integrity is crucial, subsequent profiling and improvement techniques are also required.

This paper describes a method for cleansing data retrospectively. Although this work relates to failure information being used for reliability analysis, this approach may be applied to other types of information.

2. BACKGROUND

2.1 Problem Statement

A business need to undertake CBM and reliability modelling was identified. Associated metrics require historical failure data, which is ostensibly stored in a corporate computerized maintenance management system (CMMS). Minimum data for basic reliability/CBM analysis is:

1. What component failed?
2. What was the component's failure mode?
3. What system was affected?
4. How was it affected?
5. What was the component's age at failure?

If a system is appropriately documented (e.g. by using reliability block diagrams), then by knowing the specific component failure mode it is possible to deduce answers to questions 1 to 3. This then reduces the dynamic information to: (a) system affected; (b) the component failure mode; and (c) age at failure. Item age may need to be deduced from dates of failure and return to service.

Unfortunately CMMS data is rarely complete or accurate. A qualitative analysis of the data in one CMMS database identified a number of problems that included:

- Failure/maintenance events were assigned to the wrong piece of equipment or inappropriate system. In particular, most items were recorded against one of the highest level systems (e.g. propulsion engine) rather than the specific system affected (e.g. engine lubrication system) and the repairable item (e.g. main oil pump seal) subsequently maintained.
- Little information was supplied regarding the root cause of failure and when data was entered into a 'Root Cause' field, the contents did not support other information supplied in the work order.
- Age of the asset at the time of failure/maintenance was not recorded as part of the job (although this information could be inferred from data stored in another CMMS table that contained data obtained from periodically recording hours).
- Parts consumed were not always assigned to the work order.
- System defaults were misused.

Although these deficiencies were not objectively quantified (techniques for which are currently being developed by the authors), the severity of the problem became obvious when the data was analysed. In this case, estimates for failure distributions become nonsensical; they simply did not match users' experiences in the field (e.g. an asset would be deemed inoperable for most of the past 2 years, which clearly had not been the case).

A comprehensive program for improving the quality of data in the client's CMMS is currently being developed, however this can only ever improve the quality of future data. A number of reasons were identified for improving the quality of historical CMMS work order data:

(1) Waiting until enough 'clean' work histories are obtained to ensure robust analysis would necessitate delaying reliability and CBM programs for several years, which was considered unacceptable.

(2) A considerable amount of data was (or potentially could be) contained in legacy systems that would never be recaptured by the organisation.

(3) Data related to assets early in life; this could never be replicated, yet could provide invaluable insight into deterioration mechanisms as well identifying reliability growth or decline.

(4) Effecting change in existing maintenance processes (such as improving the quality of data capture) is hard enough, trying to do so without being able to demonstrate how that data can be, and will be, used is virtually impossible.

(5) Using the data and identifying relationships between quality of outputs and inputs facilitates the development of robust business cases for funding implementation of the data quality enhancement program.

Consequently, a project was initiated to determine failure modes could be assigned to work orders retrospectively, and if so, to develop the tools and/or process to implement these improvements.

2.2 Hypothesis and Project Aim

Every day, maintenance engineers and technicians make educated guesses as to the cause of a particular equipment fault and what its rectification strategy should be. They do this by collating human experience (theirs and that of others), associated data (e.g. operating conditions), anecdotal evidence, and externally sourced information (e.g. maintenance manuals) to narrow down the list of options until the most likely failure mode and its consequent course of action are identified. It was therefore hypothesised that:

(a) The same approach would be appropriate for determining failure modes retrospectively.

(b) Provided enough information could be collated to sufficiently reduce the number of likely failure modes, a significant number of previously un-coded or miscoded failure modes could be deduced with reasonable confidence.

(c) The results of RCM/FMEA/FMECA analyses could provide the necessary information to limit the number of possible failure modes relevant to a particular failure event (i.e. work order).

It is important to recognise that the aim of the project was not to automatically assign failure modes to all work orders, but to develop a tool that a reliability engineer (or equivalent) could use to help 'fill in the gaps' with reasonable guesses, thereby improving the quality of information available for reliability modelling. The authors acknowledge and accept that guessing is not as good as 'knowing for sure', but this is almost impossible long after the event and may even be difficult at the time of failure. However, even an improvement from poor to average is still an improvement. Whether that improvement is worth the effort of achieving it, obviously depends on the effort and the extent of the improvement.

2.3 Case based reasoning

A paradigm commonly used to mimic human inductive reasoning is case-based reasoning (CBR). It is a methodology for solving new problems by adapting the solutions of similar past problems [1]. CBR is founded on two key hypotheses: (1) similar problems have similar solutions; and (2) the types of problems experienced in the past are likely to reoccur. These do not however prevent the use of CBR for solving quite different problems, which is done using flexible retrieval and adaptation methods, but it does imply that a significant number of problems are merely variants of earlier experiences.

In computing terms, CBR is a four stage process that:

(1) retrieves the most relevant cases to a particular problem from a databank of previous problems and solutions;

(2) determines the 'fit' between the problem and best retrieved option (using clustering techniques);

(3) adapts the solution accordingly; and finally,

(4) adds the successful adaptation to the case-bank for future use.

Although it is generally referred to as a form of artificial intelligence, case-based reasoning is actually a methodology that usually (but not always) uses artificial intelligence techniques for its implementation [3]. Traditionally, case-based reasoning was considered incompatible with data stored in relational database management systems (e.g. most CMMS). More recently this has been shown not be the case, especially if CBR is considered a methodology rather than a technology [4-6].

One difficulty in applying CBR is deciding what characteristics should be used to define the case, and thus how best to adapt previous situations to new problems. Most implementations use rules; however these can be difficult to define in advance. Consequently, the automated process of adaptation is often replaced with a human user. These systems "exploit the memory processes developed in CBR research, while relying on a human user to adapt and evaluate solutions" [7-10]. They are sometimes differentiated from fully computerized CBR approaches by calling them memory-based reasoning or case-based aiding systems.

Unfortunately, cases described in natural language text are not natural CBR domains [7] as meanings are difficult to match because of the number of possible language variants; thus significant effort is required to distil the information into a usable form prior to being synthesised into cases.

Finally, both a strength and a weakness of CBR is that it readily accepts anecdotal evidence into its information base. This is particularly useful when there is insufficient hard data to identify statistically significant solutions beforehand. Additionally, case-based reasoning can be used with a very small initial case-base, which expands as experience and the number of problems identified increases. Thus it is slowly being applied to Asset Management problems, where it is often too difficult or cumbersome to develop enough generalised solutions for all likely problems and their solutions.

2.4 RCM/FMEA Analysis

RCM and FMEA are proactive but subjective processes for defining aspects of asset management policy relating to the systems and their components being investigated. They are very good at helping businesses understand the effects of potential failure modes of assets in their specific operating contexts. Consequently, the results of well executed studies should be invaluable repositories of information. In particular, they contain data pertaining to how failures can be distinguished, what maintenance should be performed for a particular failure mode and the criticality of a particular failure mode, information that should be very useful to everyday maintenance and operations activities. Similarly, this data could be useful for helping to retrospectively interpret the thoughts and actions of maintainers and operators as recorded in CMMS failure histories (work orders), which are generally used for initiating maintenance to rectify failures.

Unfortunately, utilising RCM/FMEA data in this manner is not as simple as it may seem. Reasons for this, as perceived by the authors and published in [11, 12], primarily relate to differences in emphasis between RCM/FMEA processes and general maintenance activity. Examples include:

- RCM/FMEA reports information pertaining to functions and their failures, whilst work orders and failure records typically report observable events and actions performed.
- A significant proportion of RCM/FMEA data refers to fault prevention and system testing (e.g. corrective actions and controls) that can be used to prevent failure. This information is of limited use in developing cases for diagnosing failures that have already occurred.
- Although the list of RCM/FMEA failure modes should correspond with the failure modes identified in the field, this is rarely the case ([11] reported only a 20% overlap). The completeness of the RCM/FMEA database depends largely on the expertise of team that performed the initial study.
- Each RCM/FMEA record refers to a distinct root cause in a given component/part. However the technician is generally only interested in the failure of major functions of the unit (e.g. injector failure). These may have multiple underlying failure modes or root causes (e.g. nozzle blockage, spring failure etc) with the same operational or observable effect (e.g. lack of injection). Unless the required course of action for the technician/operator is different separating these failure modes has little operational benefit.
- RCM/FMEA records usually contain failure effects and/or failure modes not observable to the technician and therefore of little use in fault finding or deciding what maintenance to perform.
- RCM/FMEA studies are often focussed on failures that effect safety, environmental protection and/or critical components, whilst most work orders are focused on operational performance and the severity of its impact.
- Different technical jargon is used in FMEA studies to that supplied in work orders.
- Control systems are not commonly analysed, yet cause a significant number of operating problems.

Additional issues relating to the textual and subjective nature of RCM data further complicates its utilisation for ongoing reliability analysis:

- Answers to all RCM/FMEA questions are generally stored as free text and therefore the language/terminology used depends on the person entering the data.
- Functional location 'bit' hierarchies used in a CMMS system may be very different to the subjective system-subsystems defined by a RCM/FMEA study.
- There is no generic taxonomy used for defining and classifying assets and their components.
- No classification or codification schema for functions, failures, effects or failure modes. This makes comparing aggregating failure information across systems and/or studies virtually impossible.

These also complicate its utilisation for case-based reasoning. Many of these issues have been overcome in the solution described herein.

3. WHAT IT DOES

A search tool, called 'RCM Failure Finder', has been developed to find likely candidates for failure modes that correspond to a particular CMMS failure record, based largely on the information in the work order. This set of potential failures, approximately analogous to a case-bank, was developed from RCM data. Previous history records were not included in the initial version of the search engine as data therein was not deemed sufficiently accurate. (Strictly speaking, this set should not be called a case-bank as it is not based on "what happened before"; instead it uses examples of "what could happen".)

Searching is initiated by a user from a screen which concurrently displays original and, when completed, cleansed data associated with a CMMS work order (See Figure 1). To initiate a new search for failure modes that may be relevant to the current work order, the user presses a button. A search string is automatically created by appending information from various fields in the work order as shown in Figure 1, Step 1, which is immediately tokenised (described shortly) and compared against the RCM case-bank.

On a new screen, the user views a list of possible failure modes ordered by their calculated relevance, as shown in Figure 1, Step 2. The full RCM record associated with any element in the list can be viewed on the same screen (Figure 1, Step 3). Note that in addition to the usual RCM information, the client's data base also contains information about the observable effects of each failure. This proves to be very useful for finding the 'right' failure modes. The user then selects the appropriate failure mode and returns to the first screen. If no suitable record can be found in the list, the user can:

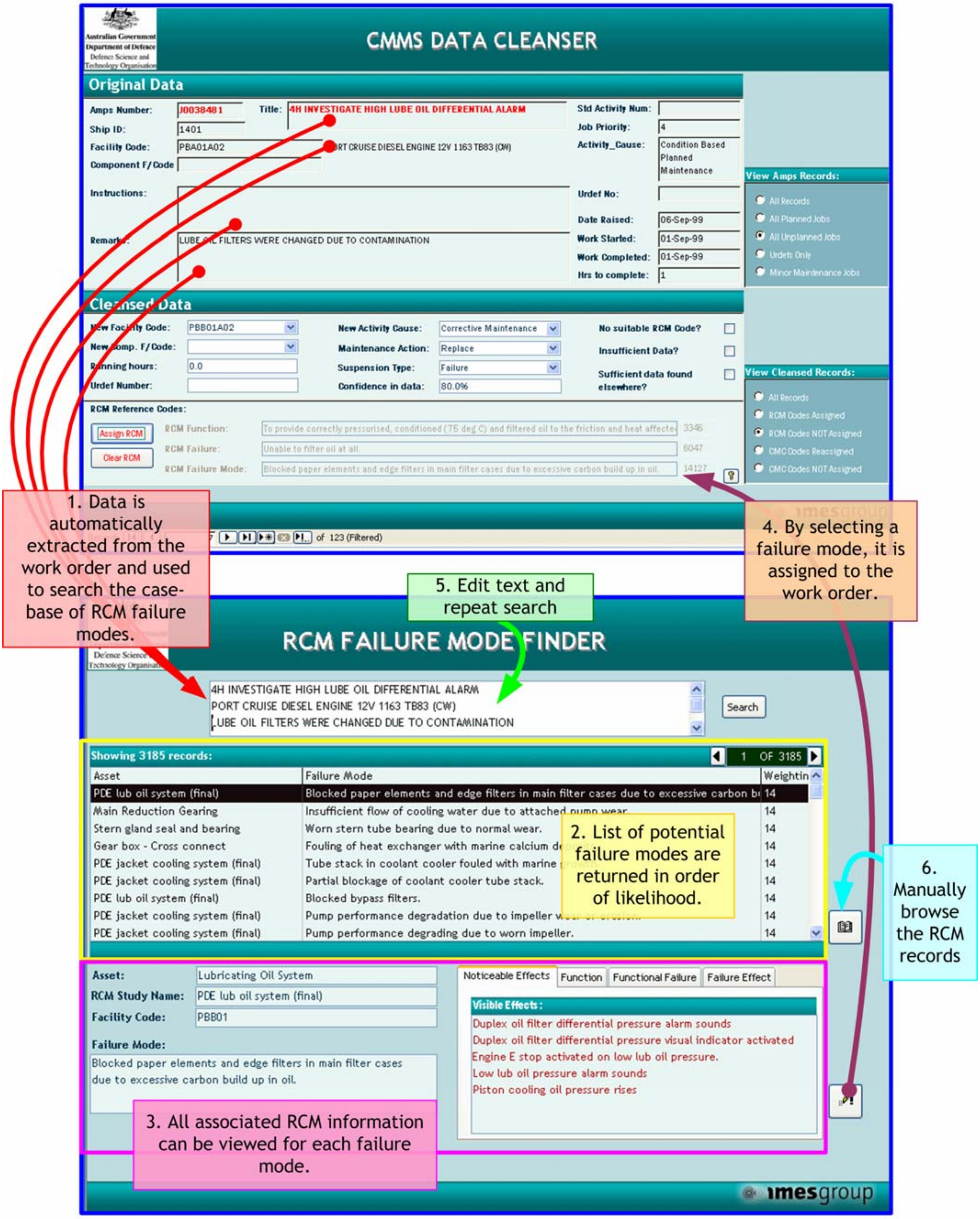

Figure 1: Screenshots of CMMS data viewer and RCM Failure Finder

(a) Amend the search text and repeat the search (Figure 1 step 5), making it possible to incorporate externally sourced data that may be missing from, or erroneous in, the original work order.

(b) Explore the case-bank manually (Figure 1 step 6). This action is automatically logged by the tool, so that improvements to searching functionality can be implemented,

(c) Flag the CMMS work order as containing insufficient information for failure mode assignment. (This usually occurs if there is more than one RCM failure mode that shares the same observable effects that have been reported in the work order and no external information is readily available.)

(d) Flag that no suitable RCM failure modes can be found for the CMMS work order (i.e. the RCM study did not identify the particular failure mode, or perceive it as likely.)

The tool also allows the user to append or change other work order information following failure mode identification. In some cases this is done automatically (e.g. assignment of the appropriate facility code), although any such assignments can be easily overridden by the user. He/she can also assign a confidence level against the new data. Unfortunately, due to the customer's reticence to change any historical data in their CMMS, a separate database of "clean" history records is being managed in the interim. Although not optimal, other systems developed allow it to be a functional work-around until process issues can be resolved.

Results of data cleansing are presented via a web-based data portal. Records can then be downloaded (currently in Excel, Word and XML formats) or directly plotted by other users for analysis or modelling.

4. SYSTEM DEVELOPMENT AND EXECUTION

System development was composed of several key steps:

- Developing an interface for displaying related data from disparate engineering data sources;
- Determining what work order and RCM data should be used comparison (token definition);
- Converting the RCM data records into a searchable case-bank;
- Developing mapping between RCM and CMMS equipment taxonomies;
- Writing scripts to execute the search functions.

Each of these will now be described separately.

4.1 Aligning Disparate and/or Segregated Data

Asset information is contained in a number of separate systems that are often difficult to interrelate. In addition to the CMMS and RCM data, associated information was also kept in a variety of databases and document management systems. Some of this data was required to help identify the failure mode associated with a particular work order.

In order to facilitate this and other data centric projects, a system was developed to present information captured by a wide variety of enterprise data-producing systems, by means of a convenient intranet-based interface. By contextualising the data, only relevant information from the underlying data sources is displayed.

Relevant to this paper, this interface facilitated access to live RCM and CMMS data, as well as allowing the user of 'RCM Failure Finder' to source information pertaining to the failure event that is contained in other systems. Results were also displayed using the same system (as this became merely another data source).

4.2 Defining Search Tokens

Tables of unique keywords and key phrases for use in case-bank development and case retrieval were derived using the following process:

1. Commonly used words and phrases were extracted in the RCM database, CMMS static tables, past CMMS work order text descriptions and ISO14224 taxonomy using basic SQL queries.
2. Determiners, pronouns, prepositions and conjunctions were removed, whilst nouns, adjectives, verbs and adverbs were retained.
3. Non-technical nouns were removed by comparing the contents of the list with a free online engineering dictionary.
4. Acronyms (including commonly used codes) were identified. Their expanded forms or meanings were defined.

5. Roots of individual words were identified and ‘non-acronyms’ were assigned with a unique word root (e.g. propulsion, propeller, propelled etc were all assigned to ‘prop’).
6. Common spelling errors or multiple spellings were identified and assigned a consistent spelling.
7. Slang terms were identified with a list of possible meanings (e.g. ‘stuffed’ could mean broken, cracked, corroded, inoperable etc).
8. Related adjectives and adverbs were linked (e.g. higher/lower, above/below etc).
9. Common groupings of keywords were then identified as key phrases.

Items were classified so that different types could be managed differently during program execution. Groups were defined as keywords, key phrases, acronyms, synonyms, place names (e.g. site names/ship compartments), slang and spelling. Collectively, these items will be referred to as tokens.

4.3 Building the case-base (search domain)

Initial data cleansing involved fixing spelling mistakes in static database table fields (both RCM and CMMS databases) such as failure mode/effect descriptions, system and subsystem names. Errors in CMMS work orders were not rectified, as these tables are dynamic and thus continually being populated with new data.

Using simple scripts, each RCM record was parsed to identify any tokens in its text fields (e.g. asset descriptions, functions, failure modes, failure effects, visible effects etc). The list of all tokens found was stored as one single string that was then assigned to the relevant RCM failure mode. Multiple instances of the same token were retained. The case-bank therefore consists of a table of these tokenised RCM strings and the failure mode reference to which they belong.

An example case string for a blocked engine lube oil filter is given in Figure 2.

ANZAC SHIP:PLATFORM SYSTEMS:PROPULSION:PROPULSION DIESEL ENGINE: LUBRICATING OIL SYSTEM:UNABLE FILTER OIL: BLOCKED PAPER ELEMENTS EDGE FILTERS MAIN FILTER CASES EXCESSIVE CARBON BUILD UP OIL OIL FILTER PAPER ELEMENTS BLOCKED DIFFERENTIAL PRESSURE VALVE OPERATE SOUNDING ALARM MACHINE CONTROL ROOM PAPER ELEMENTS BLOCK RELIEF OPEN BYPASS FILTERS ACTIVATING VISUAL DIFFERENTIAL PRESSURE INDICATOR EDGE FILTERS BLOCK ENGINE LOW LUBE OIL PRESSURE ALARM SOUND PISTON COOLING OIL SUPPLY INDICATION SHOW APPARENT INCREASE SENSOR UPSTREAM FILTERS EMERGENCY STOP ACTIVATE PRESSURE FALLS BAR WATCHKEEPER INTERVENE ALARMS EMERGENCY STOP OCCUR CHANGE FILTERS SHIP LIMITED ECO GT ENGINE OUT OF SERVICE DUPLEX OIL FILTER DIFFERENTIAL PRESSURE ALARM SOUNDS DUPLEX OIL FILTER DIFFERENTIAL PRESSURE VISUAL INDICATOR ACTIVATED ENGINE E STOP ACTIVATED ON LOW LUB OIL PRESSURE LOW LUB OIL PRESSURE ALARM SOUNDS PISTON COOLING OIL PRESSURE RISES

Figure 2: Example of a tokenised case

Due to the issues described in Section 2.4, preparing the case-bank was the most time-consuming part of the development process. Whilst initial generation was automated, some fine tuning was required, mainly to adjust cases that contained terms that may lead to erroneous results. However, once prepared, ongoing maintenance has been minimal. This primarily involves ‘tweaking’ the list of tokens and adding new RCM records to the case-bank.

4.4 Equipment taxonomy

Specific problems with the particular CMMS being analysed necessitated a new structure. The functional location hierarchy was not well defined and only listed systems, subsystems and a small selection of components. As an example, the CMMS system could associate a propulsion engine lube oil filter failure as:

HMAS Anzac (1401)
- Propulsion (P)
  - Diesel Propulsion Systems (PB)
    - Lube oil systems (PBB01)
      - STBD Diesel Engine Lube Oil System
      - PORT Diesel Engine Lube Oil System

The only maintainable items listed as part of the Starboard/Port Lube oil systems were ‘Oil pump’ and ‘Straight stop valve’. With this structure, a blocked lube oil filter failure could only be assigned to the Port/Stbd lube oil system.

ISO Taxonomy Element	Description	Example
Industry:	Type of main industry	Defence
Business Category:	The type of business or processing stream	Maritime Systems
Installation Category :	The type of facility	Major Surface Ships
Plant/Unit Category:	The type of Plant/Unit	HMAS Anzac
Section/System:	Main Section/System of the plant	Propulsion System
Equipment (Class/Unit):	Class of similar equipment units. Each equipment class contains equal equipment units (e.g. Compressors).	Propulsion Diesel Engines
Subunit:	A subsystem necessary for the equipment unit to function.	Lube oil system
Component/Maintainable Item:	The group of parts of the equipment unit that are commonly maintained (repaired/restored) as a whole.	Duplex lube oil filter
Part:	A single equipment item.	Filter cartridge

Table 1: An example of how the ISO 14224 taxonomy can be applied

In comparison, an RCM process could classify the same failure as:

Asset: Anzac (Class)
- Platform Systems
  - Propulsion
    - Propulsion Diesel Engine
      - Lubricating oil system
        - Function
          - Functional Failure
            - Failure mode (blocked lube oil filter)

Differences between these RCM and CMMS functional location structures were overcome by creating a new equipment hierarchy, based on the taxonomy presented in ISO14224 [13], and mapping this to both the RCM failure modes and CMMS equipment structures separately (see Figure 3). A generic structure was applied so that once a hierarchy for a particular type of equipment was developed the same structure could be used across different systems and subsystems. It was also not easy to map hierarchies directly. An example of how ISO14224 categorises the same failure mode is shown in Table 1.

Using a richer hierarchy allowed failures to be more accurately and consistently assigned to a system/sub-system/unit. Lists of equipment, components and parts were developed from: (a) the RCM system/subsystem breakdown; (b) manufacturer's documentation; and (c) industry best practise. These were kept as generic as possible to facilitate reuse and easy comparison across similar equipment types (e.g. diesel engines used for propulsion and generation). Another benefit of this approach is that it facilitates comparison of data pertaining to multiple instances of the same subunits, maintainable items and/or parts.

It is important to emphasise that the authors are not suggesting that this schema should be adopted by CMMS and RCM

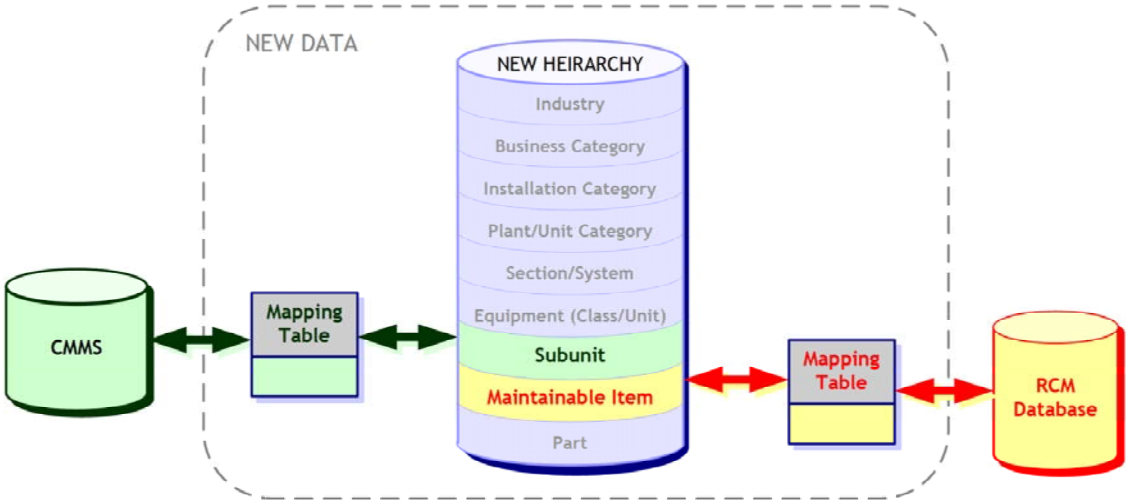

Figure 3: New hierarchy designed in accordance with ISO14224 is used to link the CMMS and RCM equipment lists.

systems. In fact, they embrace diversity of bit hierarchies, of which there should be as many as required to analyse the data easily and appropriately. This ensures that analysts are not constrained by the format in which the data is stored or was initial set up. Providing that mapping tables are complete, a 'bit' in one schema can be converted to its respective 'bit' in any another schema. Different users can thus view the same data in ways most applicable to them.

4.5 Executing a search

Case retrieval uses search functions that were developed using a mixture of Visual Basic and SQL. The process of executing a search for potentially relevant failure modes is shown in Figure 4. A string is first created from the work order. Superfluous words and characters (e.g. punctuation, numbers) are removed from the string. Tokens are then identified, actioned and results of these actions appended to a token list. Examples of actions include: expansion of acronyms, replacement of keywords with their word root, identifying all predefined synonyms and replacing commonly misspelt words with their correct spelling.

For each token in the token list, cases containing that token are retrieved. Each retrieved case is assigned a weighting based on the type of token used for its retrieval. Identical cases are then grouped and their weightings added. Finally, the list of unique cases is presented in order of decreasing (summated) weight.

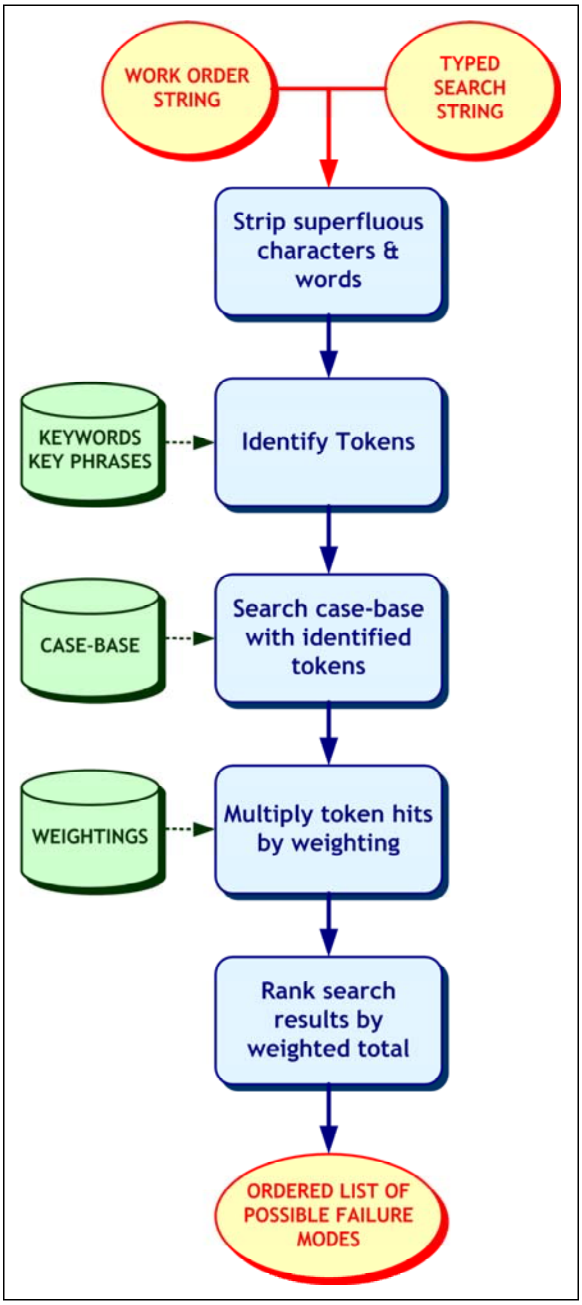

Figure 4: Case retrieval process

5. RESULTS AND DISCUSSION

This system facilitates the assignment of previously unrecorded failure modes with relative ease. On average, failure mode assignment takes between 1 and 2 minutes per work order. Delays are only incurred if information needs to be sourced from other systems, or if the work order text fields contain a significant amount of repeated information. The inclusion of 'observable effects' in the RCM database (and subsequent case-bank) improved retrieval success substantially.

Results of a typical analysis performed on 298 work orders related to the propulsion diesel engine aboard HMAS Anzac undertaken over several years, with a RCM case-bank of 433 failure modes, are shown in Figure 5.

Case retrieval was difficult when the case-bank contained a number of failure modes with the same visible or operational effects and/or associated maintenance actions. Consider the example of a work order that reports a 'failed diesel engine injector'. This CMMS record also recorded observable effects as being 'High exhaust gas temperatures' and commented that 'changing the injector rectified the problem'. The RCM database contained 8 separate injector failure modes with these symptoms. In this plant however, faulty injectors are always replaced in their entirety, so the maintainer rarely determines what internal component of the injector rendered it inoperable. Thus no information was (or will typically be) recorded as to the actual root cause of an injector failure. In such cases, the user can elect to use an example failure mode (e.g. 'injector failure - normal wear') for all similar failures unless information is available to assign otherwise. It is recognised that this is not an ideal solution; the work-around does however facilitate some failure grouping (and hence reliability analysis) that would not be possible otherwise.

Identification of failure modes was also not possible when required failure modes had not been included in the original RCM analysis, either because the entire system had not been analysed (e.g. control systems), or because the failure mode was overlooked or considered unlikely. Fortunately, as these cases are identified they can be added to case-bank, thus minimising the effect over time.

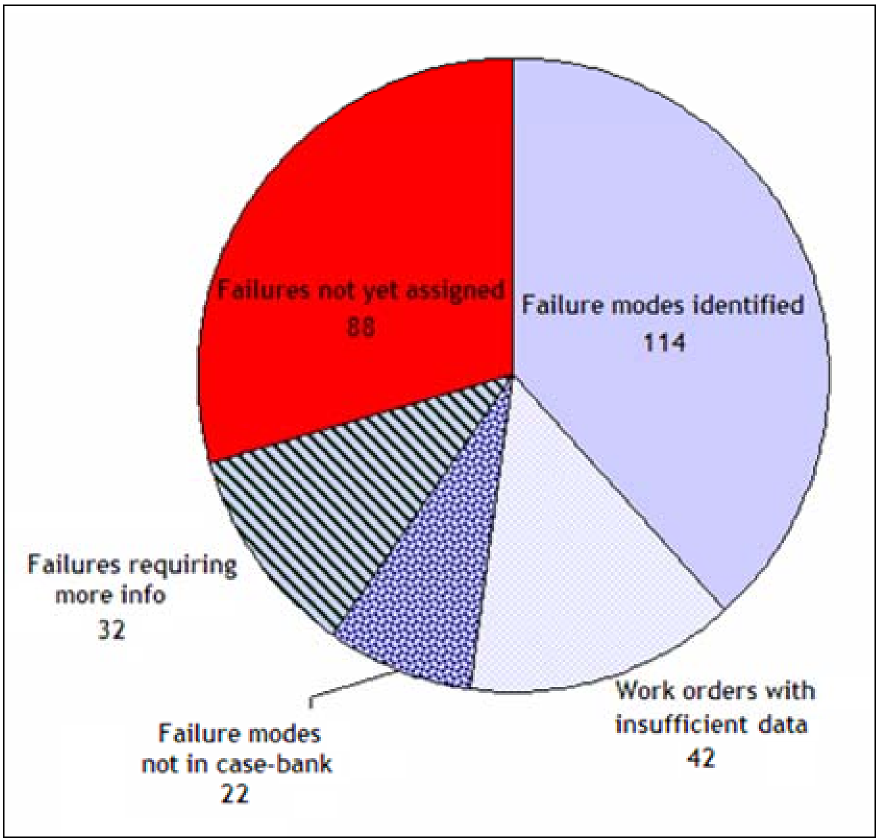

Figure 5: Distribution of typical results

6. ONGOING DEVELOPMENT AND POTENTIAL ENHANCEMENTS

This tool and process is still under development. Current improvements to case retrieval performance, retrieval speed and improving functionality include: (a) ongoing enhancement of keywords so that the most appropriate failure modes are at the top of the retrieved list more often; (b) utilizing free text functions in SQL server 2005; (c) changing from a database delivered system to an intranet-based tool and incorporating it into an existing reporting portal developed by the authors; (d) adapting the tool for troubleshooting failures in the field (i.e. at time of failure) as shown in Figure 6, first presented in [14] and (e) adding new RCM studies to the case-bank (e.g. control system, missing failure modes).

It must be emphasised that this work was undertaken using a pre-existing RCM study and CMMS. Future RCM studies and CMMS development can incorporate the wisdoms gleaned from the development of this tool and further simplify (or eliminate) the development of a new case-bank. These changes would also improve subsequent failure mode retrieval and accuracy of failure mode assignment. In particular these include:

- Considering failure modes that result in operational problems (major or minor) as comprehensively as safety/environmental impacts. (FMECA may therefore be more appropriate for developing a case-based reasoning.)
- For maintainable items that have numerous failure modes attributable to faults of their component parts (e.g. injector failures caused by blocked nozzles, broken springs, worn spring ends, wear of nozzle needle etc) aggregate failure mode records will be developed for the maintainable items in their entirety.
- Incorporating generic ISO14224 taxonomy into the standard RCM process.
- Improving consistency of language used in recording RCM results.
- Separating visible effects from internal effects that are difficult for the maintainer/operator to identify.
- Providing generic failure modes at the maintainable item level.
- Analysing Control systems by RCM as these are commonly the largest group of unassignable failure modes. They also seem to be the most difficult to troubleshoot and therefore offer the greatest benefit to the maintainer in using the data on a daily basis.
- Incorporating specific name of alarms, alerts (e.g. LH2 light on panel 4H7 turns red) in both the RCM study (and subsequent keyword list).

7. CONCLUSIONS

Although the system may seem complex, once established the case-bank is very easy to use and extend. For a new asset, it is simply a matter of cleaning the RCM data, running scripts to automatically identify new tokens and prepare the case-bank (some manual adjustment may be required), before coding new mappings between the between ISO taxonomy and any new facility codes.

Retrospective failure mode assignment initially depends on the quality of the RCM study, but as the case-based approach facilitates ongoing improvement to the case-bank (which effectively adds missing failure modes to the RCM study), this limitation decreases over time.

This paper clearly shows that it is possible to use the RCM and CMMS data "as is" in an ongoing manner and improve its usefulness over time as well as improve the quality of historical work orders that can then be used for improved reliability and CBM analysis. More generally the authors have shown that it is possible to improve the quality of data retrospectively and it is not necessary to wait for perfect data to improve business decision-making.

8. REFERENCES

1 Hodkiewicz M., Kelly P., Sikorska J. & Gouws L. (2006) A Framework to Assess Data Quality for Reliability Variables. In Proceedings of the 1st World Congress of Engineering Asset Management, Gold Coast Australia, 21/1-21/11, CIEAM.

2 Sporleder C., van Erp M., Porcelijn T. & van den Bosch A. (2006) Correcting 'Wrong-Column' Errors in Text Databases. In Proceedings of the Annual Machine Learning Conference of Belgium and the Netherlands, Ghent Belgium, 49-56.

3 Watson I. (1999) Case-Based Reasoning Is a Methodology Not a Technology. Knowledge-Based Systems, 12, 303-308.

4 Anick P. & Simoudis E. (1993) Technical Report Ss-93-07, Case-Based Reasoning and Information Retrieval: Exploring Opportunities for Technology Sharing. In Proceedings of the AAAI Spring Symposium on Case-Based Reasoning and Information Retrieval, Menlo Park, AAAI Press.

5 Rissland E. & Daniels J. (1995) *Using CBR to Drive IR*, in Thirteenth International Joint Conference on Artificial Intelligence: San Francisco

6 Shimazu H., Kitano H. & Shibata A. (1993) Retrieving Cases from Relational Databases: Another Stride Towards Corporate-Wide Case-Base Systems. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence.

7 Leake D. (1996) Cbr in Context: The Present and Future. In Proceedings of the Case-Based Reasoning: Experiences, Lessons and Future Directions, San Francisco, AAAI Press.

8 Kolodner J. (1991) Improving Human Decision Making through Case-Based Decision Aiding. The AI Magazine, 12(2), 52-68.

9 Stanfill C. & Waltz D. (1986) Towards Memory-Based Reasoning. Communications of the ACM, 29(12), 1213-1228.

10 Waltz D. (1989) Is Indexing Used for Retrieval? In Proceedings of the DARPA Case-based Reasoning Workshop, San Mateo, Morgan Kaufmann.

11 Atamer A. (2004) Comparison of Fmea and Field-Experience for a Turbo-Fan Engine with Application to Case-Based Reasoning. In Proceedings of the IEEE Aerospace Conference, IEEE.

12 Roberge P.R., Tullmin M.A.A. & Tretheway K.R. (1996) Failure Analysis by Case-Based Reasoning. In Proceedings of the 1st Global Internet Corrosion Conference, Available at: http://corrosionsource.com/events/intercorr/techsess/papers/session7/abstracts/roberge.html.

13 International Standards Organisation (2006) *Iso14224-2006: Petroleum, Petrochemical and Natural Gas Industries - Collection and Exchange of Reliability and Maintenance Data for Equipment*, 170 pages.

14 Sikorska J. & Hodkiewicz M. (2006) Fmea and Reliability Analysis: Closing the Loop between Theory and Practice. In Proceedings of the Engineers Australia Mechanical Panel Conference, Perth.

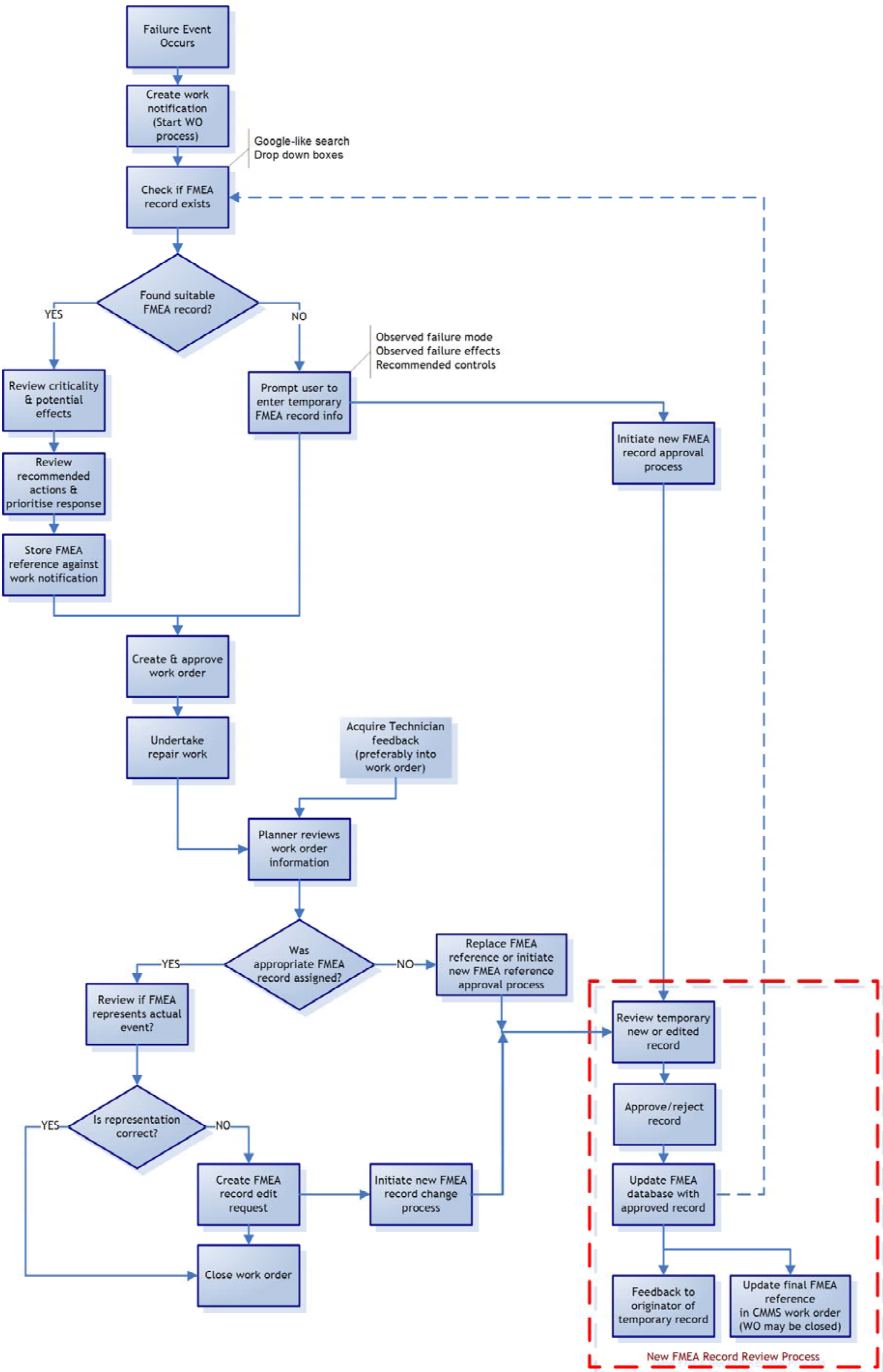

Figure 6: Integration of FMEA into maintenance and CMMS work order data entry process