Linking Product Functionality to Historic Failures to Improve Failure Analysis in Design

ROBERT B. STONE, PH.D., ASSOCIATE PROFESSOR, DEPARTMENT OF INTERDISCIPLINARY DESIGN ENGINEERING, UNIVERSITY OF MISSOURI-ROLLA
102A Basic Engineering Building, Rolla MO, 65409
rstone @umr.edu, Phone - 573.341-4086 Fax – 573.341.6593

IREM Y. TUMER, PH.D., RESEARCH SCIENTIST, COMPUTATIONAL SCIENCES DIVISION, NASA AMES RESEARCH CENTER
Moffett Field, CA 94035
itumer@mail.arc.nasa.gov

MICHAEL E. STOCK, GRADUATE RESEARCH ASSISTANT, DEPARTMENT OF MECHANICAL ENGINEERING, UNIVERSITY OF MISSOURI-ROLLA

ABSTRACT

Prior research has shown that similar failure modes occur within products (or components) with similar functionality. To capitalize on this finding, a knowledge base-driven failure analysis tool, the Function-Failure Design Method (FFDM), has been developed to allow designers to perform FMEA-style failure analysis during conceptual design. The FFDM can offer substantial improvements to the design process since it enhances failure analysis thus giving it the ability to reduce the number of necessary redesigns.

The FFDM however is only as good as the knowledge base that it draws from, and one fundamental question that arises in using FFDM is: At what level of detail should functional descriptions of components be encoded? This paper explores two approaches to populating a knowledge base with actual failure occurrence information from Bell 206 helicopters. Results indicate that encoding failure data using more detailed functional models allows for a more robust knowledge base. Interestingly however, when applying FFDM, high level descriptions continue to produce useful results when using the knowledge base generated from the detailed functional models.

Keywords: failure analysis, conceptual design, functional modeling, failure modes

1. INTRODUCTION

In most design cases it is necessary that the designer have a wide knowledge of the nature of their new design in order to develop creative and robust ways to embody the

functionality of a new product. In other words, the designer must have a useful intellectual knowledge base from which to draw concepts and evaluate them, or perform an exhaustive review of potential concepts from external sources. Knowledge base driven design methods lessen the need for a designer to have a broad and deep expertise by searching and reusing archived design knowledge. The Function-Failure Design Method (Stock, et al., 2003) provides designers a methodology for performing failure analysis in conceptual design and also aids them by using a function-based concept generator approach (Strawbridge, et al., 2002) to streamline the design process. FFDM is a start-to-finish design method that utilizes knowledge bases that link product function to likely failure modes and product function to possible concepts in order to minimize the designer's need for a large intellectual knowledge base.

The FFDM is a structured formulation of the function-failure analysis method of Tumer and Stone (2003). This method archives historical failure knowledge by linking it to functional representations of the failed component in matrix form. To accomplish this, the functional basis (Hirtz, et al., 2002) and a failure mode taxonomy (Arunajadai, et al., 2002) are used to ensure a retrievable method of archival. However, it is possible to archive this information at multiple levels of abstraction. This paper investigates the process of populating function-failure knowledge bases at two such levels of abstraction in hopes of arriving at a reusable and robust methodology that can be applied to a wide range of engineering designs.

In order to provide background on failure prevention in product design, this paper begins with a review of the prevalent methods for performing failure analysis on new designs in Section 2, with special attention given to the function-failure method of Tumer and Stone (2003) and FFDM (Stock, et al., 2003). Since these two methods are rooted in functional modeling, an explanation of the various levels of functional modeling is also given in this section. Two methods for populating a knowledge base for use in FFDM are given in Section 3 along with the presentation of two sample knowledge bases. These knowledge bases are compared and used in an FFDM design case in Section 4. The paper finishes with conclusions and future work in Section 5.

Before using the FFDM methods in a design case, it is necessary to develop a knowledge base to link failure modes to the functions that they are likely to occur for. With this in mind, it is equally important to keep these two processes separate in order to better understand the material being presented here. Section 3 covers the generation of such a

knowledge base. After generating a knowledge base, it could be used by many different designers when faced with a wide range of design cases, one such case is shown in Section 4. However, it is not necessary that those designers using a knowledge base in a design case have populated the knowledge base, the process of knowledge base population is time-consuming, but a good knowledge base could be applied to many disparate design problems.

2. BACKGROUND

2.1 Current Failure Analysis Methods

Several failure analysis methods currently exist and are used in industry, but by far the most widely used method is Failure Mode and Effect Analysis (FMEA). FMEA is a widely used method because it can be applied to systems, processes and product designs (Stamatis, 1995). In this paper, our review emphasis is placed on failure analysis for product design. FMEA was originally developed by the U.S. Military (MIL-P-1629A, 1980) and its methods have been refined by different industries since its inception (AIAG, 1993). Even with this process refinement and formalization, there still exists multiple shortcomings within the failure analysis of FMEA. These shortcomings include a lack of well-defined terms (Lee, 1999), problems with identifying key failures (Bednarz and Marriott, 1988) and subjective analyses based on the user's experience (Bell, et al., 1992). Another common complaint of the FMEA process is that it is tedious (Hunt, et al., 1995) and that engineers consider it to be "laborious" (Wirth, et al., 1996).

When concerned with product design, it is important that failure analysis is performed early in the design process in order to reduce the necessary amount of redesigns. McKinney (1991) underlines the importance of performing failure analysis in conceptual design, but goes on to report that FMEA is commonly performed too late in the design cycle and has very little effect on the overall product design. To improve on the "classical" FMEA methods numerous attempts have been made to apply failure analysis during conceptual design. The FLAME system (Hunt, et al., 1995;Price, 1996) applies a computer simulated analysis to electrical system functional representations early in the design cycle. The FLAME system is a well-documented success of conceptual failure analysis but is limited to electrical systems.

In system design, the Advanced FMEA (AFMEA) method of Kmenta et al. (1999) can be used to perform failure analysis on a functional representation of a system design. Much

like FLAME, AFMEA seeks to capitalize on fewer physical redesigns by addressing possible failures before concrete physical representations of the design have been developed. Successful attempts at conceptual product design failure analysis are however much harder to come by. The CFMA method of Hari and Weiss (1999) is one such method, but has shortcomings in that it actually assumes some degree of product form, thus making it not truly "conceptual."

To achieve a failure analysis method that is suitable for actual conceptual design implementation, it appears that the most applicable methods are those that rely on knowledge bases to alert the designer of possible failure modes within their new design. Knowledge base failure analysis methods began with the early matrix techniques for FMEA logistical archiving (Barbour, 1977;Collins, et al., 1976;Goddard and Dussault, 1984). The WIFA system (Wirth, et al., 1996) is a similar system that populates knowledge bases with information from past failure analyses. This information is archived using standardized languages in order to improve the comprehensibility and reusability of failure analyses. The WIFA (a German acronym for "knowledge-based FMEA") system is similar to the function-failure analysis method of Tumer and Stone (2003), with the exceptions of application stage and the theory behind failure mode enumeration. In WIFA, the analysis is performed within the traditional FMEA timeframe, which has been previously noted as being "too late" in the design cycle to truly guide and improve the design. To combat this, Tumer and Stone tailored their method for use in conceptual design. Also, in WIFA the failures are enumerated for system elements but in the function-failure method, this is not possible. Since it is applied in the conceptual stage, Tumer and Stone's method cannot rely on system elements since their physical form is unknown and products only exist as functional representations. Therefore, the function-failure analysis methods base their failure mode enumeration methods strictly on the desired functionality of the product being designed.

2.2 The Function-Failure Design Method (FFDM)

FFDM is a methodology that allows designers to perform failure analysis in conceptual design (Stock, et al., 2003). The method is advantageous to a designer because following its steps can possibly reduce the number of necessary redesigns, thus shortening the overall design cycle. FFDM allows designers to use a wealth of archivied knowledge from historical failure occurrences and analyses to guide their new designs. FFDM is suitable for use in new design or redesign and is well-suited for use with the concept

generator methods of Strawbridge et al. (2002). A graphical representation of FFDM format can be seen in Figure 1.

As shown in Figure 1, FFDM requires a knowledge base of historical failure occurrences linked to product function in order to generate the likely failure modes for new designs. This knowledge base is generated using the method shown in the work of Roberts et al. (2002). This process of population relies on a user to develop two matrices that will then be multiplied together to produce a third matrix, which will be known as the function-failure knowledge base. This process begins by acquiring historical failure knowledge on an artifact. The type of failure is classified within the failure mode vocabulary of Arunajadai et al. (2002) and is then related to the artifact within the component-failure (**CF**) matrix. Within **CF**, the rows represent artifacts and the columns represent failure modes. A numerical value of '1' present in cell **CF**$_{ij}$ indicates that the j-th failure mode occurred for the i-th artifact. Upon completing the **CF** matrix, functional models are developed for each failed artifact and are also entered into matrix form. The function-component (**EC**) matrix contains i sub-functions as row entries and j artifacts (or components) as column entries. As before, a value of '1' in **EC**$_{ij}$ indicates that the j-th artifact exhibited the i-th sub-function within its functional representation. The function-failure (**EF**) matrix is generated by multiplying **EC** and **CF** together. This matrix relates historical failure occurrences to functionality and is used as the knowledge base in the FFDM approach.

Step 1: Determine black-box model, complete with high-level function and flow pairing

Step 2: Use black-box function and flow pairing to query EF knowledge base for possible failure modes

Failure mode 1
Failure mode 2
Failure mode 3
...

Step 3: Develop detailed functional model to accomplish desired product functionality while best addressing possible failure modes from step 2

Concept Generator Approach

Traditional Design Approach

$$F \cdot X = M$$

Step 4a: Use concept generator approach to arrive at possible product solutions

Step 4b: Enumerate concept variants to satisfy the functions within the detailed functional model

CF

specific failures/
component

Step 5a: Use the component-failure (CF) matrix to evaluate each product solution

Step 5b: Evaluate concept variants with respect to the possible failure modes from step 2

Step 6: Select component solution or concept variant that best addresses the historically common failure modes
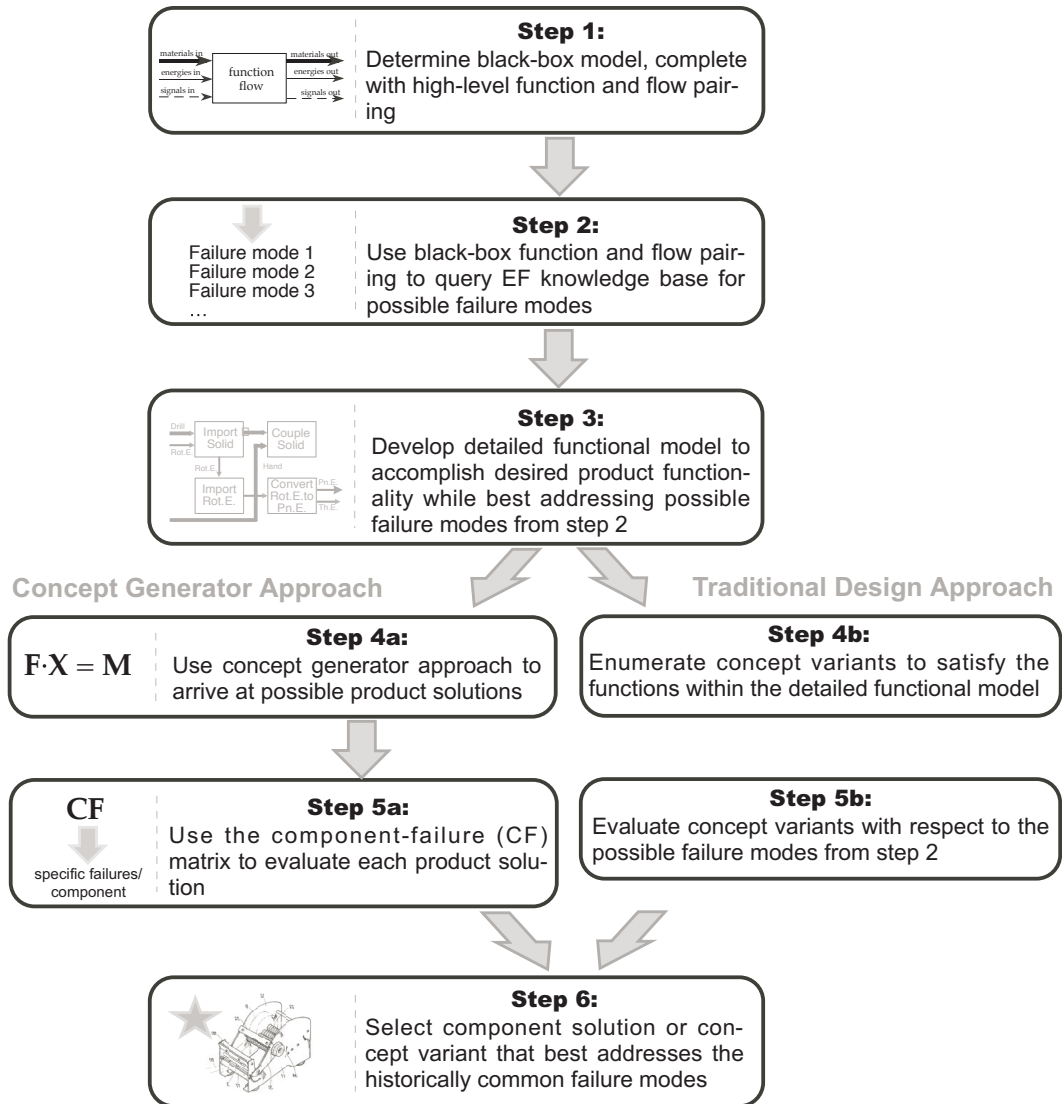
Figure 1. The FFDM Procedure.

When developing a knowledge base that can be applied across a wide range of design domains and applied to many different designs, it is important to use standardized vocabularies to archive information within the knowledge base. Utilizing standardized vocabularies limits ambiguity between different users and also maintains a serviceable size for the knowledge base. In other words, standardized vocabularies ensure that multiple entries of the same failure mode or function do not reappear under other aliases. The standardized vocabularies for failure modes and functionality used within FFDM also benefit the user by supplying exhaustive definitions of the terms within them (Arunajadai, et al., 2002;Hirtz, et al., 2002). These standardized vocablariess are also necessary for in the matrix computations shown later and also serve to limit the size of the matrices.

The concept generator (Strawbridge, et al., 2002) is an approach that embodies a functional model with concepts that it draws from a knowledge base known as the function-component matrix (**X**). Since FFDM guides designers to the formulation of functional models, the concept generator can be easily integrated into the design process. In a similar manner to the **EF** knowledge base, the **X** matrix is developed by investigating many products and relating the components within them to the functions that they perform. This is accomplished by generating functional models for the given artifact and then "reverse engineering" it to determine which of its components embody each function within the functional model. This method also takes advantage of the functional basis (Hirtz, et al., 2002) by using its vocabulary to archive within, and query from, the **X** matrix.

Since they both rely on functional representations, the function-failure knowledge base and the concept generator are used in conjunction within FFDM. FFDM first generates a list of likely failure modes based on a very high-level functional description of a new design by querying the function-failure knowledge base (Steps 1 and 2 in Figure 1). A more detailed functional model is then developed (Step 3) and the concept generator uses this functional model to enumerate possible concept variants (Step 4a). (More traditional concept generation techniques can be used in Steps 4b and 5b.) These concepts are then evaluated based on the list of possible failure modes (Step 5a) to arrive at a design that best addresses the historical likelihood of failure occurrence within the new product.

2.3 Functional Modeling

Functional models are graphical representations of product (or component) functionality (Otto and Wood, 2001). Functional models can be developed for existing products, but offer great benefits when they are linked with the design process to represent desired product functionality in order to satisfy customer needs. Functional models have been shown to provide a basis for organizing the design process, enhance creativity in design and allow designers to generate more solutions. Overall, functional modeling is a useful tool in developing successful products from the conceptual design stage.

However, functional models can exist at many different levels of abstraction (Gietka, et al., 2002) and it is important to know the uses and limitations of each level. Since FFDM requires the use of functional modeling at multiple levels of abstraction, a rigorous definition of the levels is given here. Verma and Wood (2003) propose three levels of functional modeling based upon the level of product detail contained within the model itself. These levels are enumerated as the black box, the design and the reverse

engineering level of functional modeling. As expected, the black box level defines and represents only the most basic functionality and flows contained within the product or design. In this work, we define the black box level to contain only one sub-function and its associated flows. The design and reverse engineering levels are similar and are therefore the hardest to discern between. A design level functional model represents an initially detailed representation of the sub-functions that act on the multiple flows that pass through the product being analyzed. This level leaves some amount of abstraction within the model and is most useful in conceptual design, thus garnering its name. The reverse engineering level is the most detailed model of the system and gets its name because these models are usually constructed after "tearing down" a product and analyzing each of its components. These different levels can be seen for an electronic scale in Figure 2.
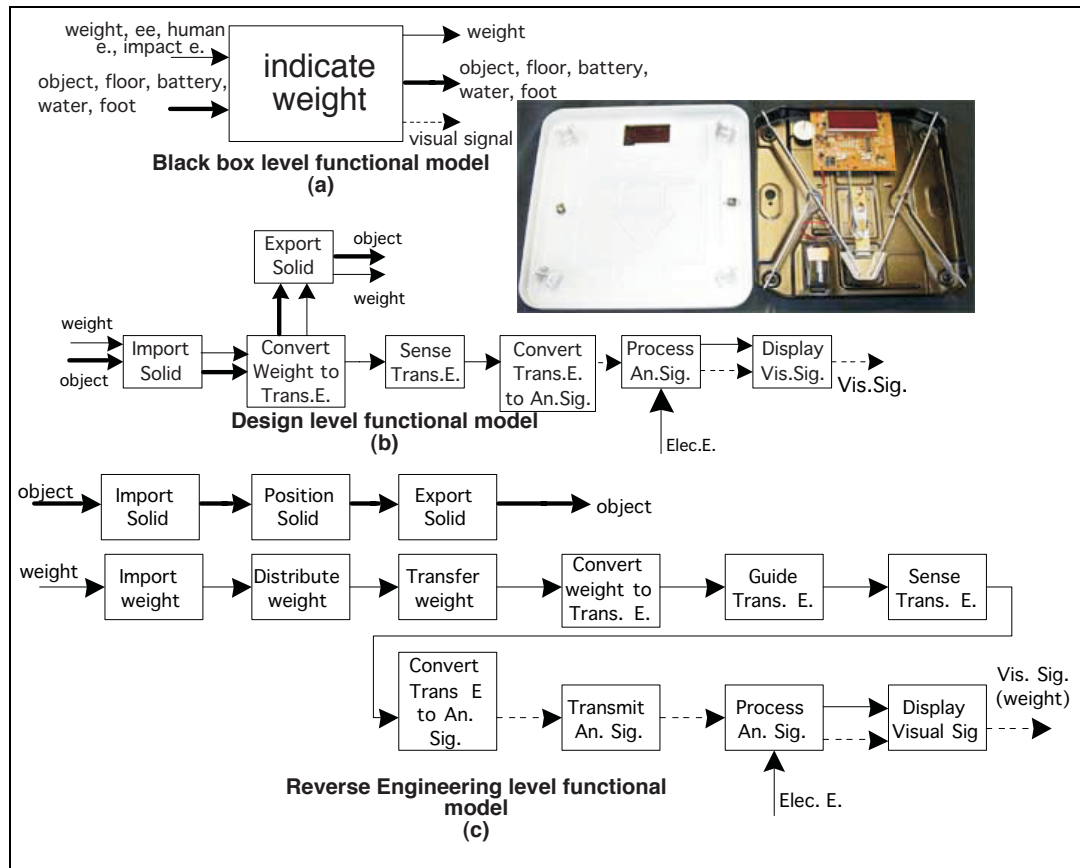

Figure 2. Different Levels of Functional Modeling for an Electronic Scale.

Figure 2(a) shows the black box level functional model of the scale. In this functional model, only the overall product function of 'indicate weight' and incoming and outgoing

flows are shown. Figure 2(b) shows a fragment of the design level functional model including the input flows of weight and object and the output flow of visual signal. The design level functional model exhibits some amount of form independence and represents an intermediate level of modeling between the vague black box level and the most detailed reverse engineering level. Finally, Figure 2(c) shows the reverse engineering level functional model for the same flows as in Figure 2(b). At this level, the functionality of the actual components guides the derivation of the functional model. It can be seen that the reverse engineering functional model takes the sub-functions of the design functional model to a more detailed level to express the functionality of the actual components within the design product.

Each of these levels of functional modeling is important within the design process, especially when taking advantage of design information reuse and design by analogy methods. FFDM takes advantage of two forms of design information reuse by reusing past concepts from the concept generator and past failures from the function-failure knowledge base to guide its design process. Therefore, since multiple levels of functional modeling are used within FFDM, it is imperative to have a good understanding of the difference between them. The concept generator relies on a knowledge base of historical product designs to develop new concepts. This knowledge base, known as the **X** matrix, is developed by constructing **reverse engineering** level functional models for multiple products, linking the sub-functions from the model to components within the product and storing this information in matrix form. When used to generate concept variants, the concept generator can accept either **design** or **reverse engineering** level models for a new design.

On the other hand, FFDM strives to use a new product's functionality from its **black box** level functional models to develop an initial list of likely failure modes that the product will exhibit. However, the fundamental question arises whether the historical knowledge used to populate the function-failure knowledge base should be encoded at the **black box** or the **reverse engineering** level. We show that the concept generator (Strawbridge et al., 2002) allows for knowledge to be encoded at one level of functional modeling and queried at a less detailed level. Is this possible in FFDM? This gives rise to the one fundamental concern of populating the function failure knowledge base: Since it is desired to use FFDM at the black box level for new designs, should actual component failures be linked to the components' black box level function or should they be linked to more detailed component functionality?

3. METHODS FOR POPULATING FUNCTION-FAILURE KNOWLEDGE BASES

3.1 Initial Efforts

Roberts et al. (2002) constructed the first function-failure knowledge base by collecting failure information on Bell 206 rotorcraft using National Transportation Safety Board (NTSB) accident reports. Components' failures were determined from these reports and functional models were developed for each of the failed components. The functional models of these components varied between one and five sub-functions to describe the component. In this initial test of the function-failure analysis of Tumer and Stone (2003) the level of functional modeling did not strictly adhere to any of the aforementioned levels as described by Verma and Wood (2003). The level of functional modeling used by Roberts et al. can best be described as fitting between the black box and design levels.

Previous work by the authors (Stock, et al., 2003) used more detail in developing a function-failure knowledge base using the same failure occurrence information as Roberts et al. (2002). In this more recent effort, the authors developed a function-failure knowledge base after developing reverse engineering level functional models of the failed components within the Bell 206 helicopter. When used within the structure of FFDM, this detailed knowledge base showed improved failure analysis over FMEA.

3.2 Two Function-Failure Knowledge Bases at Distinct Levels of Detail

To determine which level of functional modeling is best suited for developing a function-failure knowledge base, an experiment is undertaken in which two knowledge bases are constructed, compared and used to perform failure analysis during the conceptual design of a new product within the FFDM framework. The first knowledge base to be constructed will utilize component functional models at the black box level, showing similarity to the method of Roberts et al.[1] (2002). This function-failure knowledge base will be referred to as **EF$_1$**. The second knowledge base (**EF$_2$**) will consist of the

---

[1] The actual Roberts et al. function failure knowledge base is not being used in this comparison because of its inconsistency in number of sub-functions per component functional model. Modeling in this fashion is ambiguous since it is difficult to determine the necessary number of sub-functions to adequately model the component.

function-failure information harvested by Stock et al. (2003). The component functional models in **EF₂** were developed at the reverse engineering level using the repeatable functional modeling methods of Kurfman et al. (2003).
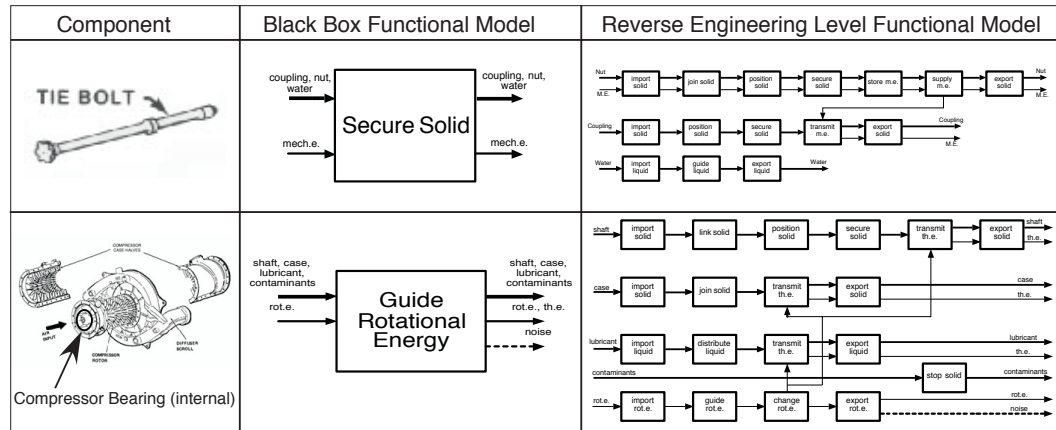

Figure 3. Functional Models Used to Populate EC_1 and EC_2.

To develop these two knowledge bases, three matrices are generated. A single component-failure matrix is generated and named **CF$_{rotorcraft}$**. This matrix contains information on 25 failed components that span seven systems within the Bell 206 rotorcraft. These systems include the compressor, engine, powertrain, turbine, airframe and the fuel and rotor systems. Multiple systems were chosen since studying systems across the entire rotorcraft makes for a knowledge base that can be applied to more diverse design problems. The 25 failed components exhibited 15 unique failure modes within the failure mode vocabulary of Arunajadai et al. (2002). These failure modes were determined by studying the NTSB reports and relating the information contained therein to the primary and secondary identifiers for the failure modes within the vocabulary (Tumer, et al., 2003). Two unique **EC** matrices are populated, **EC₁** and **EC₂**. **EC₁** is populated by relating artifacts to their black box functional representation while **EC₂** is populated by relating the same artifacts to their reverse engineering level representations. The functional models in the second column of Figure 3 represent a sample of those used to populate **EC₁** at the black box level. Similarly the models in the third column of Figure 3 show a sample of component functional models at the reverse engineering level that are used to populate **EC₂**. In doing so, **EC₁** contains only 11 unique sub-functions, while **EC₂** contains 55 unique sub-functions. This is due to the increased detail of the reverse engineering level functional models used to populate **EC₂**, these functional

models contain between five and 18 sub-functions depending on the functional complexity of the component under review. For example, the O-ring component contains only five unique sub-functions while the more complex fuel governor and tail rotor blade components necessitate 18 unique sub-functions to completely model their functionality. In contrast, the black box functional models contain just one sub-function for each component.

The function-failure knowledge bases are generated through the following operations:

$$\mathbf{EF}_1 = \mathbf{EC}_1 \times \mathbf{CF}_{rotorcraft} \qquad (1)$$

$$\mathbf{EF}_2 = \mathbf{EC}_2 \times \mathbf{CF}_{rotorcraft} \qquad (2)$$

In common terms, equation (1) is populating the function-failure knowledge base by linking each unique component failure occurrence to that component's black box functionality within the knowledge base. For example, assume that the crank handle of the meat grinder in Figure 4(a) has two failure occurrences, one occurrence of brittle fracture and one occurrence of direct chemical attack. Since the black box functionality of the crank handle is 'convert human energy to rotational energy,' a value of '1' would be added to the $\mathbf{EF_1}$ cells that relate 'convert human energy to rotational energy' to brittle fracture and to direct chemical attack. Conversely, populating the function-failure knowledge base at the reverse engineering level, as shown in equation (2), will relate component failure occurrences to every sub-function within the reverse engineering level functional model of the crank handle. In this case, the functional model of the crank handle contains 12 sub-functions as seen in Figure 4(c). Therefore, if the crank handle were entered into $\mathbf{EF_2}$, a value of '1' would be added to each of the cells relating these 12 sub-functions to brittle fracture and direct chemical attack.

Another important point to note in the derivation of $\mathbf{EF_1}$ and $\mathbf{EF_2}$ is that the component failure ($\mathbf{CF_{rotorcraft}}$) matrix is binary in data representation. That is, it contains only '0' and '1' for numerical values. This is done to ensure that one component does not unfairly skew the knowledge base simply because more failure information was available for it. For example, using the case above, if it were known that the meat grinder crank handle failed four times via brittle fracture and once via direct chemical attack, they would still both be entered into $\mathbf{CF}$ as the value '1.' Thus at this point, the number of failure occurrences per component has not entered into the function failure knowledge bases. Future work in this area involves using the number of occurrences for each failure mode to guide designers in assessing failure probability for their new design. Since the $\mathbf{CF}$ matrix is binary in data representation, it is imperative that many different components

exhibiting the same functionality be entered into it in order to avoid normalizing the knowledge contained therein.
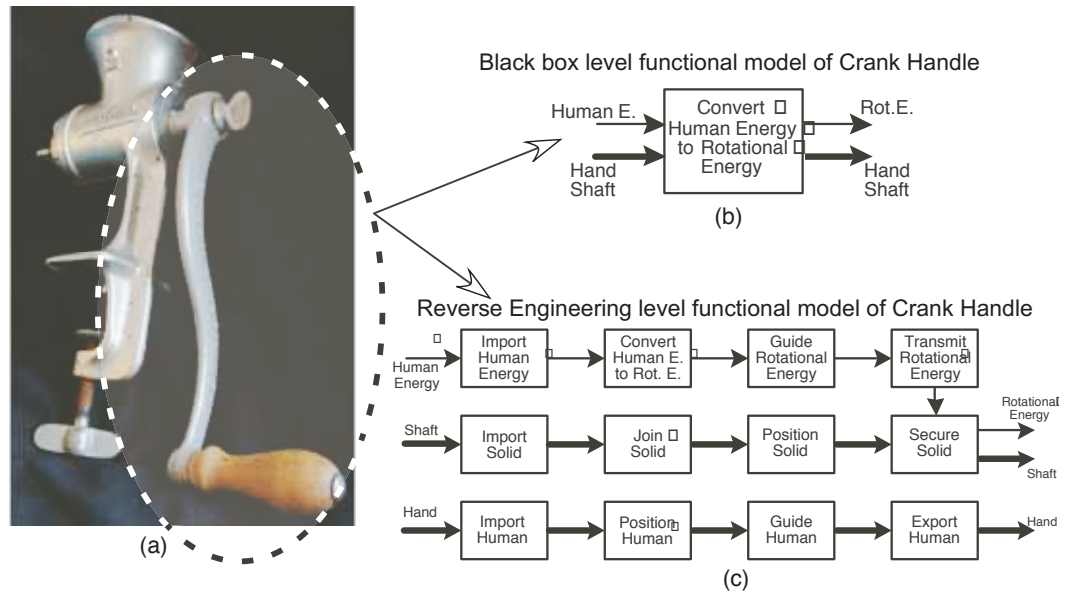

Figure 4. Functional Models of a Meat Grinder Crank Handle.

4. COMPARISON OF FUNCTION-FAILURE KNOWLEDGE BASES

4.1 EF_1 vs. EF_2

The EF_1 function-failure knowledge base can be seen in Table 1 and EF_2 can be seen in Table 2.   Upon initial examination the most glaring difference between the two knowledge bases is the fact that EF_2 contains far more sub-functions than EF_1.   This is directly related to the size of EC_1 and EC_2, as explained above.

It can be seen that EF_1 contains only eleven sub-functions to go along with the fifteen unique failure modes within the knowledge base. Knowing that many functions will be needed within the knowledge base before it can be applied to diverse design problems, it is easy to see that many more failed components within the knowledge base will be needed before this style of population will result in a knowledge base robust enough for use with FFDM. In other words, EF_1 in its current state could only be used in design cases that contained the functions within its limited scope.

Table 1. EF_1.

Function/Failure	Abrasive Wear	Adhesive Wear	Buckling	Corrosion Fatigue	Deformation Wear	Direct Chemical Attack	Force Induced Deformation	Fretting Fatigue	Galling and Seizure	High Cycle Fatigue	Low Cycle Fatigue	Stress Corrosion	Thermal Fatigue	Thermal Shock	Yielding
Change Gas	0	0	0	0	0	0	0	0	0	2	0	0	1	1	1
Convert RotE to PnE	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
Guide PnE	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Guide RotE	0	0	0	0	0	0	0	0	2	2	0	0	0	1	0
Regulate Liq	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
Secure Solid	0	0	0	0	0	1	1	0	0	2	0	0	0	1	3
Stabilize Solid	0	1	0	0	0	2	0	0	0	1	0	0	0	0	1
Stop Gas	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stop Liquid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Transmit PnE	0	0	0	0	0	0	0	1	0	2	1	0	0	0	1
Transmit RotE	1	1	0	0	1	1	0	1	0	2	0	1	1	1	0

By contrast, EF_2 exhibits 55 unique sub-functions after populating it with information from the same 25 components as EF_1. Using the same logic as before, if EF_2 was to be used in an FFDM design case, it would prove helpful for designs that could include five times the functionality as EF_1. Therefore, populating a function-failure knowledge base at the reverse engineering level of functional modeling requires fewer failed components to arrive at a more useable knowledge base. In addition, it is hypothesized that linking failure modes to every sub-function occurrence of a given function and flow pairing will yield a robust knowledge base for use in conceptual design.

It is interesting to note that even though knowledge bases EF_1 and EF_2 contain exactly the same components and failure modes, populating them at different levels of functional abstraction (the only difference in their population) can lead to greatly different representaitons of the knowledge contained therein. Even though EF_2 contains more function-flow pairings it is important not to jump to the conclusion that it is a better knowledge base simply because of this fact. To further investigate the quality of the knowledge representation in each knowledge base and to determine which is best-suited for use in the FFDM, it is important to compare the failure mode distributions within each knowledge base.

Table 2. $\mathbf{EF_2}$.

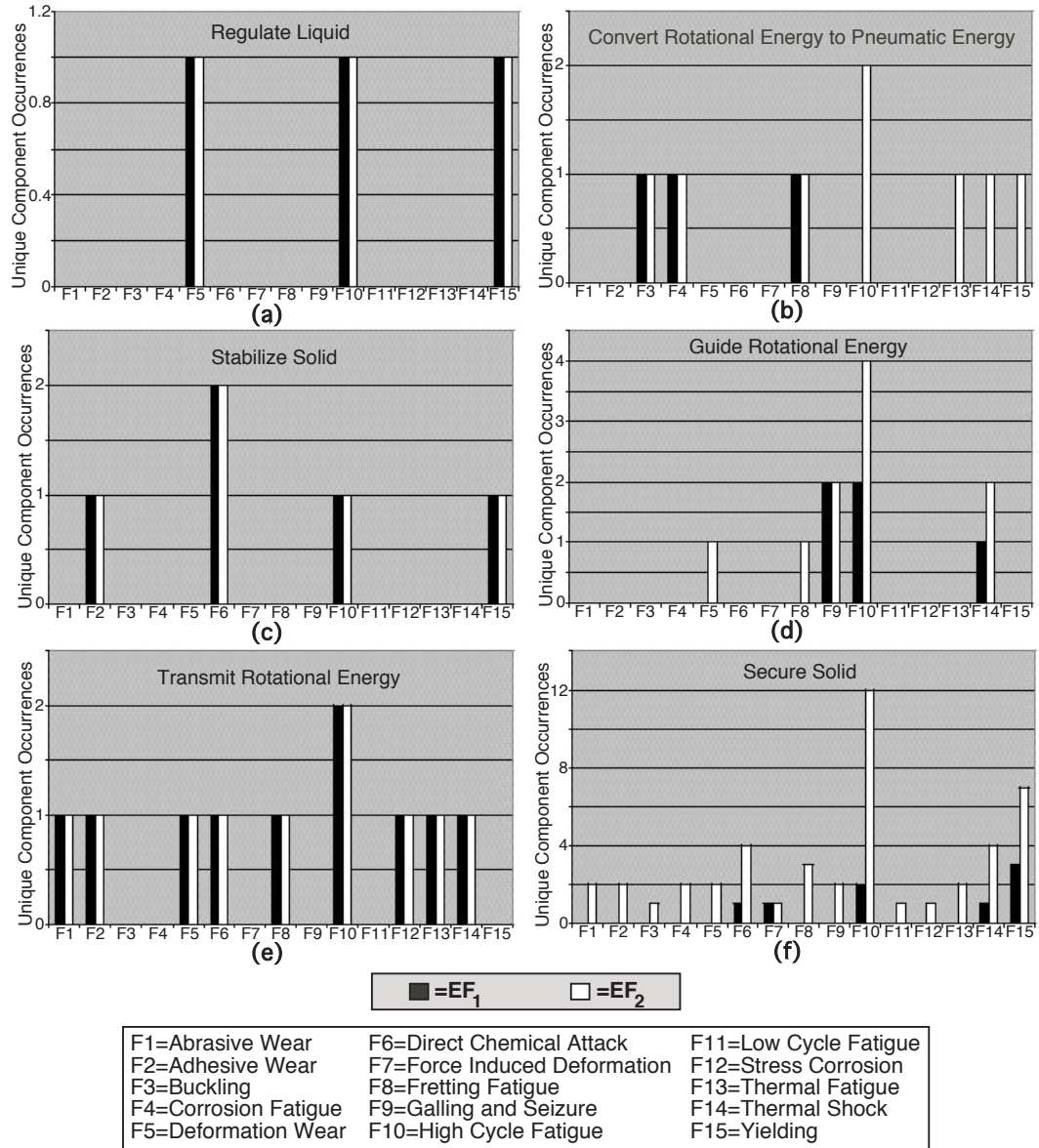| Function/Failure	Abrasive Wear	Adhesive Wear	Buckling	Corrosion Fatigue	Deformation Wear	Direct Chemical Attack	Force Induced Deformation	Fretting Fatigue	Galling and Seizure	High Cycle Fatigue	Low Cycle Fatigue	Stress Corrosion	Thermal Fatigue	Thermal Shock	Yielding
Change Gas	0	0	1	1	0	0	0	1	0	2	0	0	1	1	1
Change Liquid	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
Change PnE	0	0	0	0	0	0	0	0	0	2	0	0	1	1	1
Change RotE	0	0	0	0	1	0	0	1	2	3	0	0	0	2	0
Convert HE to RotE	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2
Convert PnE to ME	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Convert RotE to ME	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
Convert RotE to PnE	0	0	1	1	0	0	0	1	0	2	0	0	1	1	1
Couple Solid	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
Distribute Liquid	0	0	0	0	1	0	0	1	2	3	0	0	0	2	0
Distribute ME	0	2	1	1	1	3	0	2	0	7	1	0	0	1	3
Distribute ThE	2	0	0	0	1	1	0	0	0	4	0	1	2	1	2
Export Gas	2	0	1	2	0	0	0	2	0	4	1	1	2	1	2
Export HyE	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
Export Liquid	0	1	1	2	2	3	1	2	2	6	0	0	0	3	3
Export ME	0	0	0	0	0	2	1	0	0	2	0	0	0	0	2
Export PnE	0	0	1	2	0	0	0	2	0	4	1	0	1	1	2
Export RotE	1	0	0	0	0	0	0	0	2	3	0	1	1	1	0
Export Solid	2	2	1	2	2	4	1	3	2	12	1	1	2	4	8
Export ThE	1	0	0	0	1	1	0	0	0	2	0	1	1	0	1
Guide Gas	1	0	1	2	0	0	0	2	0	4	1	1	2	1	2
Guide HyE	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
Guide Liquid	0	1	1	2	1	3	1	1	0	3	0	0	0	1	3
Guide PnE	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Guide RotE	0	0	0	0	1	0	0	1	2	4	0	0	0	2	0
Guide Solid	2	0	1	2	0	0	0	1	0	3	0	1	2	2	1
Import Gas	2	0	1	2	0	0	0	2	0	4	1	1	2	1	2
Import HE	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2
Import HyE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Import Liquid	0	1	1	2	2	3	1	2	2	6	0	0	0	3	3
Import ME	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0
Import PnE	0	0	0	1	0	0	0	1	0	4	1	0	1	1	2
Import RotE	1	0	1	1	0	0	0	1	2	5	0	1	2	2	1
Import Solid	2	2	1	2	2	4	1	3	2	12	1	1	2	4	8
Import ThE	1	0	0	0	1	1	0	0	0	2	0	1	1	0	1
Inhibit Liquid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Join Solid	0	2	1	1	2	3	1	3	2	9	1	0	0	3	7
Link Solid	2	1	0	1	1	1	0	1	2	6	0	1	2	3	1
Position Solid	1	2	1	2	2	3	1	2	2	9	0	1	2	4	6
Regulate HyE	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
Regulate Liquid	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
Regulate ME	0	1	0	0	0	2	0	0	0	2	0	0	0	1	1
Secure Solid	2	2	1	2	2	4	1	3	2	12	1	1	2	4	7
Stabilize Solid	0	1	0	0	0	2	0	0	0	1	0	0	0	0	1
Stop Gas	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stop HyE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Stop Liquid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Stop PnE	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Stop Solid	0	0	0	0	0	0	0	0	2	2	0	0	0	1	0
Store ME	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3
Supply ME	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2
Transmit ME	0	1	0	1	1	2	1	1	0	5	1	0	0	1	6
Transmit PnE	0	0	0	0	0	0	0	1	0	2	1	0	0	0	1
Transmit RotE	1	1	0	0	1	1	0	1	0	2	0	1	1	1	0
Transmit ThE	2	0	0	0	2	1	0	1	2	7	0	1	2	3	2

Figure 5. Comparison of Functions Within EF_1 and EF_2.

Figure 5 shows the differences between the two existing knowledge bases, EF_1 and EF_2. For seven of the 11 functions within EF_1, the failure distribution contained therein is the same as that within EF_2. The failure mode distribution for three of these functions can be seen in Figure 5(a), (c) and (e). This behavior is the result of the given functionality appearing in the reverse engineering models for only the components for which it was in their black box model as well. On the other hand, the four sub-functions that exhibited different failure mode distributions between the two knowledge bases, 'change gas', 'convert rotational energy to pneumatic energy', 'guide rotational energy' and 'secure solid', can be found in many reverse engineering component models but not as frequently

in the less detailed black box level models. The most glaring case of this situation occurs for the sub-function 'secure solid' as seen in Figure 5(f). 'Secure solid' is the black box sub-function for only six of the failed rotorcraft components but occurs in twenty-four of the reverse engineering level functional models.

By studying Figure 5, it can be seen that certain failure modes do indeed occur more frequently for some functions. None of the sub-functions within either EF_1 or EF_2 exhibit an even distribution of failure modes. This allows a designer to use the information in the knowledge bases to predict the failure modes that are most likely to occur for their new designs based on desired product functionality. This fact can streamline the design process by ensuring that some degree of failure avoidance is designed into the initial physical representation of new design or redesign.

4.2 Using Each Knowledge Base in a New Design Case

In this section, a design problem is proposed to test the utility of EF_1 and EF_2 within FFDM. To do so, a design problem is developed that meets with the functionality present within the two knowledge bases. In this comparison, a small hand-held air compressor will be designed. This compressor should be powered by a hand held electric drill and be capable of clearing debris from an area such as a workbench. One deisgner's interpretation fo the FFDM procedures in the design of this device has previously been developed using the EF_2 knowledge base (Stock, et al., 2003). Note that this material is being repeated for completeness. This design, as well as the design methodology can be seen in Figure 6. In this product design case, using FFDM with knowledge base EF_2 led directly to the inclusion of shaft support bearings, increased heat transferring area, improved chucking interface, and a filter screen for the incoming air passage on the compressor.
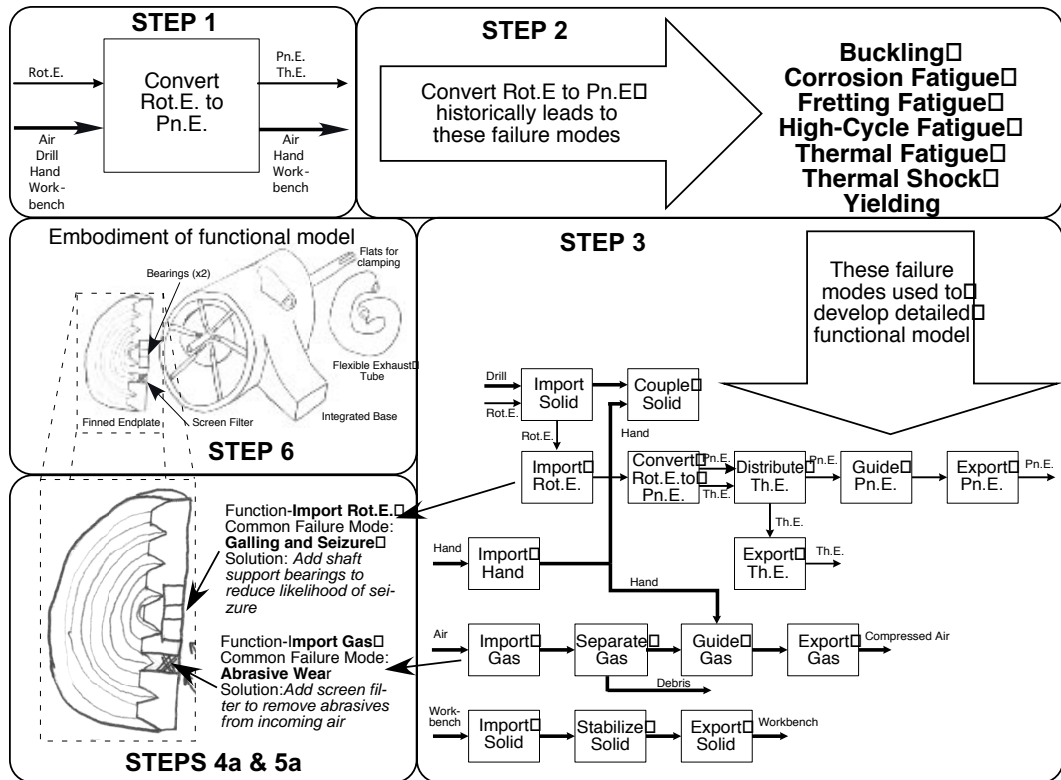
Figure 6. Compressor Design Using FFDM and EF_2.

Following the same design process with EF_1 is quite difficult and shows the inherent problems in using a knowledge base that contains only few sub-functions. When generating the list of common failure modes from the black box function 'convert rotational energy to pneumatic energy,' there are less selection criteria for possible concept variants and it appears that this detracts from the thorough failure analysis usually seen in FFDM. When using EF_1 for this task, only three possible failure modes are generated, less than half of the seven potential failure modes generated by using EF_2. Noticeably absent in the list from EF_1 is high cycle fatigue and any thermal effects. Further FFDM analysis shows that the possibilities of galling or seizing within the rotating componentry are also ignored when knowledge base EF_1 is used.

It is difficult to develop a completed design with EF_1, but it easy to note that the failure analysis would be much less thorough than if knowledge base EF_2 were used. Strictly adhering to the recommendations within FFDM leads to an overall design similar to that seen in Figure 6 but does not include shaft support bearings, incoming air filter or thermal finning. Additionally, fatigue analysis would not likely be conducted, even though it was conducted when EF_2 was used in the design case. Because of this, preliminary results show that it is preferred that future function-failure knowledge bases be developed using

methods similar to that used in the genereation of $\mathbf{EF_2}$. Generating a function-component matrix with functional models at the reverse-engineering level leads to a knowledge base that can be better used in the FFDM design process.

5. CONCLUSIONS/FUTURE WORK

The knowledge-base driven failure analysis tool improves the design process by limiting redesigns and increasing the importance of failure analysis. Methods such as FFDM can decrease the necessary time to conduct failure analyses (Stock, et al., 2003) and by moving failure analysis to conceptual design can make it more powerful and influential in product design (McKinney, 1991). However, as in all design, the strength and breadth of the user's knowledge base is the key to FFDM. A main advantage of FFDM is that the user does not need to possess a vast intellectual knowledge base. FFDM's function-failure knowledge base dictates the effectiveness of the analysis that is performed. FFDM is truly a case of being "only as good as your knowledge base." Knowing this, substantial effort has been undertaken to determine the best level of component functional model abstraction to arrive at the most robust and versatile knowledge base.

This paper has presented two approaches for populating the $\mathbf{EC}$ matrix, using a black box level of functional modeling and using a more detailed reverse engineering level of modeling. Encoding knowledge into the $\mathbf{EC}$ matrix with reverse engineering level models yields a more robust function-failure knowledge base for use within FFDM. Not only is encoding information at this level an efficient method to populate a large knowledge base, it has been shown that such a knowledge base allows for a more thorough failure analysis during conceptual design. Therefore FFDM can be used to the best of its capability in performing failure analysis in conceptual design, minimizing the need for costly and time-consuming redesigns.

Future work in the area of function-failure knowledge base population will focus on developing larger knowledge bases and applying them to disparate design cases to evaluate the utility of FFDM. It is also desired to populate similar knowledge bases with past FMEA information in order to supplement the knowledge bases that contain actual failure occurrence information. Work is also underway to evaluate this methodology by archiving design knowledge from NASA and applying FFDM techniques at the JPL Product Design Center.

ACKNOWLEDGEMENTS

REFERENCES

AIAG (1993) *Potential Failure Mode and Effects Analysis (FMEA) Reference Manual*, Automotive Industry Action Group.

Arunajadai, S. G., Stone, R. B. and Tumer, I. Y. (2002) "A Framework For Creating a Function-Based Design Tool for Failure Mode Identification," *Proceedings of the 2002 ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, DETC02/DTM-34018, Montreal, Canada.

Barbour, G. L. (1977) "Failure Modes and Effects Analysis by Matrix Methods," *Proceedings of the 1977 Annual Reliability and Maintainability Symposium*.

Bednarz, S. and Marriott, D. (1988) "Efficient Analysis for FMEA," *Proceedings of the 1988 Reliability and Maintainability Symposium*.

Bell, D., Cox, L., Jackson, S. and Schaefer, P. (1992) "Using Causal Reasoning for Automated Failure Modes and Effects Analysis (FMEA)," *Proceedings of the 1992 Reliability and Maintainability Symposium*.

Collins, J. A., Hagan, B. T. and Bratt, H. M. (1976) The Failure-Experience Matrix: A Useful Design Tool, *Journal of Engineering for Industry*, 98(3): 1074-1079.

Gietka, P., Verma, M. and Wood, W. H. (2002) "Functional Modeling, Reverse Engineering, and Design Reuse," *Proceedings of the 2002 ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, DETC2002/DTM-34019, Montreal, CA.

Goddard, P. L. and Dussault, H. B. (1984) "The Automated Matrix FMEA-A Logistics Engineering Tool," *Proceedings of the 1984 The Society of Logistics Engineers' 19th Annual Symposium*.

Hari, A. and Weiss, M. P. (1999) "CFMA-An Effective FMEA Tool for Analysis and Selection of the Concept for a New Product," *Proceedings of the 1999 ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, DETC99/DTM-8756, Las Vegas, NV.

Hirtz, J., Stone, R., McAdams, D., Szykman, S. and Wood, K. (2002) A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, *Research in Engineering Design*, 13(2): 65-82.

Hunt, J. E., Pugh, D. R. and Price, C. P. (1995) Failure Mode Effects Analysis: A Practical Application of Functional Modeling, *Applied Artificial Intelligence*, 9(1): 33-44.

Kmenta, S., Fitch, P. and Ishii, K. (1999) "Advanced Failure Modes and Effects Analysis of Complex Processes," *Proceedings of the 1999 ASME Design Engineering Technical Conference, Design for Manufacturing Conference*, DETC99/DFM-8939, Las Vegas, NV.

Kurfman, M. A., Rajan, J., Stock, M. E., Stone, R. B. and Wood, K. L. (2003) Experimental Sudies Assessing the Repeatability of a Functional Modeling Derivation Method, Accepted to *Journal of Mechanical Design*.

Lee, B. H. (1999) "Design FMEA for Mechatronic Systems Using Bayesian Network Causal Models," *Proceedings of the 1999 Design Engineering Technical Conferences, Design Automation Conference*, DETC-99/DAC-8605, Las Vegas, NV.

McKinney, B. T. (1991) "FMECA, The Right Way," *Proceedings of the 1991 Reliability and Maintainability Symposium.*

MIL-P-1629A (1980) *Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, United States Department of Defense.

Otto, K. and Wood, K. (2001) *Product Design: Techniques in Reverse Engineering, Systematic Design, and New Product Development*, Prentice-Hall, New York.

Price, C. P. (1996) "Effortless Incremental Design FMEA," *Proceedings of the 1996 Annual Reliability and Maintainability Symposium.*

Roberts, R. A., Stone, R. B. and Tumer, I. Y. (2002) "Deriving Function-Failure Information for Failure-Free Rotorcraft Component Design," *Proceedings of the 2002 ASME Design Engineering Technical Conferences, Design for Manufacturing Conference*, DETC2002/DFM-34166, Montreal, Canada.

Stamatis, D. H. (1995) *Failure Mode and Effect Analysis, FMEA from Theory to Execution*, Milwaukee, WI, ASQ Quality Press.

Stock, M., Stone, R. and Tumer, I. Y. (2003) "Going Back in Time to Improve Design: The Function-Failure Design Method," *Proceedings of the 2003 ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, DETC2003/DTM-48638, Chicago, IL.

Strawbridge, Z., McAdams, D. A. and Stone, R. B. (2002) "A Computational Approach to Conceptual Design," *Proceedings of the 2002 ASME Design Engineering Technical Conference, Design Theory and Methodology Conference*, DETC02/DTM-34001, Montreal, Canada.

Tumer, I. Y. and Stone, R. B. (2003) Mapping Function to Failure Mode During Component Deveopment, *Research in Engineering Design*, 14(1): 25-33.

Tumer, I. Y., Stone, R. B. and Bell, D. G. (2003) "Requirements for a Failure Mode Taxonomy for use in Conceptual Design," *Proceedings of the 2003 International Conference on Engineering Design*, ICED03-1612, Stockholm, Sweden.

Verma, M. and Wood, W. H. (2003) "Functional Modeling: Toward a Common Language for Design and Reverse Engineering," *Proceedings of the 2003 ASME Design Engineering Technical Conferences*, DETC2003/DTM-48660, Chicago, IL.

Wirth, R., Berthold, B., Kramer, A. and Peter, G. (1996) Knowledge-based Support Analysis for the Analysis of Failure Modes and Effects, *Engineering Applications of Artificial Intelligence*, 9(3): 219-229.