6-1-2011

Warranty prediction during product development: Developing an event generation engine in an engineer-to-order environment

Hee-Rak Kang

Warranty Prediction During Product Development: Developing an Event Generation Engine in an Engineer-To-Order Environment

By

Hee-Rak Kang

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in Industrial Engineering

Supervised by

Professor Dr. Marcos Esterman

Department of Industrial and Systems Engineering

Kate Gleason College of Engineering

Rochester Institute of Technology

Rochester, New York

June 2011

COMMITTEE MEMBERS:

Dr. Marcos Esterman
Associate Professor, Industrial and Systems Engineering Department
Rochester Institute of Technology

Dr. Brian Thorn
Associate Professor, Industrial and Systems Engineering Department
Rochester Institute of Technology

Dr. Carol Romanowski
Associate Professor, Center for Multidisciplinary Studies
Rochester Institute of Technology

Approved By:

____

Dr. Marcos Esterman Date

____

Dr. Brian Thorn Date

____

Dr. Carol Romanowski Date

Thesis Release Permission Form

Rochester Institute of Technology

Kate Gleason College of Engineering

Title:

Warranty Prediction During Product Development:
Developing an Event Generation Engine in an
Engineer-To-Order Environment

I, Hee-Rak Kang, hereby grant permission to theWallace Memorial Library to reproduce my thesis in whole or part.

Hee-Rak Kang Date

DEDICATION

This thesis is dedicated to my grandfather, a man who after the Korean War was able to build a successful business and support his family. We buried you on a hill overlooking a place you could never go back to.

ACKNOWLEGMENTS

I must profusely thank my thesis advisors, Dr. Esterman and Dr. Romanowski. You guys spent many hours helping me figure out what to do! This work would not have been possible without both of you. Thanks to Dr. Thorn for helping me improve the readability of my thesis. It's one thing to do the research, an entirely different thing to put it down on paper. Professor Esterman, I always knew I was a little bit weird and slightly off center. I didn't realize I had company.

Going back to school after working for a few years in industry was not easy for me. Thanks to my friends, they made this transition easy. To my wife, I will always be grateful for your support even though the 1 and a half hour drive back from Rochester to Syracuse did not make me the most pleasant person to deal with. To my family, your love and unconditional support is what continues to drive me. Your continued belief in my success is a welcome respite amongst life's challenges.

ABSTRACT

In order for manufacturing companies to stay competitive, it is necessary to drive warranty system improvements in terms of improved product reliability, improved service delivery efficiency and properly designed warranty policies. However, traditional methods for assessing warranty performance are not always sufficient to alert product development teams of the impending warranty issues. Furthermore, improved assessment methods are needed to aid product development teams make decisions related to the warranty performance of the product.

The focus of this research was to develop a framework to integrate statistical inference methods and data mining techniques to create a warranty event generation framework. This was done on the context of an engineer-to-order product development environment. The objectives of this work were: (1) to develop an inference model for the integration of disparate data sources; (2) to demonstrate that multiple data streams can be conditioned for input into the above inference model; (3) to develop the above model and process in light of actual data. This thesis will report on the progress and challenges that have been made toward fulfilling these objectives. The thesis closes by outlining the future research agenda for developing a warranty event generation engine that can integrate data from disparate data sources.

TABLE OF CONTENTS

1. INTRODUCTION

Competition between manufacturers in a global marketplace has resulted in companies looking for ways to lower product cost and increase profit margins. With total warranty costs approaching $8 billion in the computer and related high tech US based companies (Mueller, 2007), controlling warranty costs during product development is one promising method for companies to gain an edge over their competition.

In order to understand how to control warranty costs, one must understand that a warranty is a contractual obligation between a consumer and manufacturer that protects the consumer should the product fail to perform its intended function within a given time period (Esterman, Gerst, Stiebitz, & Ishii, 2005). Warranty is frequently used as a marketing tool and showcases the willingness of a company to stand behind its products and services. A good example of this is Hyundai's car advertisements showcasing its "100,000 mile power train warranty" as "America's best warranty". But warranty costs are not limited to product failure in the field. They can be a result of inefficient system delivery methods or poorly designed warranty policies (Esterman, et al., 2005). One major contributor to warranty costs is getting the concept right early on in the development phase (Wilson, 1993).

Therefore, it is in the company's best interest to make any necessary design changes early on in the development process. This is because changes made later on in the development phase not only add costs but time to the product development life cycle. A delayed launch date negatively affects the total amount of revenue that can be generated during the product's life cycle. The challenge is that it is difficult to utilize incomplete and disparate data at the beginning phases of the product development.

In order to better understand this opportunity, this research looked at how companies may be able to better manage the information that they already have, such as prior distribution/historical data, quantitative data, and engineering, by utilizing a Bayesian approach. Using this type of approach allows companies to be able to handle data with small sample sizes as well as data that changes or grows over time (Campbell, 2006). This type of data integration may become helpful in assessing product reliability in the future. Current research looks at failure data at various test stages during product development in order to predict reliability growth (Mazzuchi & Soyer, 1993). This type of research focuses on reducing component or product failure modes but does not address other possible warranty events that may occur such as misaligned customer expectations.

This highlights the need for improved assessment methods to aid product development teams make decisions related to the warranty performance of the product. As a result, one of the immediate research goals was to develop a framework to integrate statistical inference methods and data mining

techniques to create a warranty event generation framework. This was done in an engineer-to-order product development environment. Engineering to Order (ETO) firms face different challenges than that of firms that respond to consumer demand. Products that are made in ETO environments tend to be complex, technologically intensive, highly specialized, capital intensive and high value (Rahim, et. al, 2003). The variety of customers that they have to cater to results in products that are at different stages of development which makes development a difficult task. Further complicating this task is that customers tend to impose their own product development process on the ETO firm (Kumar, et. al, 2009).

In addition, characteristics of the ETO system itself increase the likelihood of warranty issues such as increased system complexity and a lengthened product life-cycle. From an organizational perspective the dependency on a similar product and the high involvement of manufacturing in design would decrease the likelihood of warranty issues. This is because increased knowledge sharing would reduce the learning curve. These characteristics and differences from market based firms indicate that the ETO environment would be a good candidate for developing a warranty prediction framework.

2. PROBLEM STATEMENT

In order to address many of the challenges faced by companies to manage warranty performance during product development in an ETO environment, the main focus of this thesis was to develop methods to condition the available data streams for use in a Bayesian framework. Characteristics of ETO systems such as increased system complexity and long product life-cycle increase the likelihood of warranty issues that makes it a good candidate for testing the feasibility of this type of framework.

Traditional (non-Bayes approaches) reliability tools vary in their degree and effectiveness of predicting warranty events, and are generally used to characterize product reliability at a particular development stage. By the time field data has been collected and reviewed, the product development team is already at work on the next product revision. In order to close the gap between the time data is available to the time when this data is needed, the Bayesian framework was used to integrate field data and any available product development data. This enabled warranty performance data to be available at the any stage of the product development.

There are many challenges to this approach. One problem is the conditioning of datasets into probabilities for a Bayesian framework: prior probability, conditional probability and marginal probability. Since valuable information can be lost when data is preprocessed to fit to a distribution, combining different sources has been seen as a solution to avoid making too many assumptions and create an accurate representation of the data. Another issue with combining multiple data streams is the variety of ways sources can vary in form from point estimates, probability distributions, ratios, to qualitative. An input-output model has been provided below as a conceptual and mathematical framework (Yadav, et al., 2002).

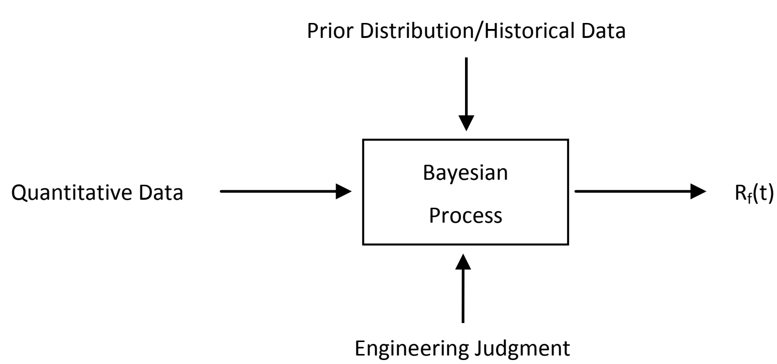

Figure 1. Bayesian Input-Output Model (Yadav & Prakash, 2002)

While information about warranty performance would be highly valuable and informative by itself, its usefulness would be extended if this estimate is updated progressively as more information

becomes available. If this process is adapted to current product development systems, development teams will be able to review predicted warranty performance at each development stage and have the capability to not only see what the current performance levels are but what they are likely to achieve. Therefore, a focus of this thesis will be to not only create an initial Bayesian model that integrates disparate datasets, but also to show feasibility that this model can be updated with new information.

The problem of combining information sources for a Bayesian framework is crucial to achieving an accurate and trustworthy warranty cost prediction model. Accelerated design process and improved cost estimation accuracy relies on the ability to combine multiple data streams into valuable information and insights. This research will seek to answer the following questions: Can multiple data streams be conditioned for input using the Bayes' Theorem? What are some of the issues that can occur when qualitative data is integrated with quantitative data? What are the different methods to condition available data streams in a Bayesian framework? These are questions that have not been answered in current literature in regards to integrating data in a Bayesian framework for warranty prediction.

Therefore, the objectives of this work were: (1) to develop an inference model for the integration of disparate data sources; (2) to demonstrate that multiple data streams can be conditioned for input into the above inference model; (3) to develop the above model and process in light of actual data. The remainder of this thesis is organized as follows: literature review of the tools used in the research, methodology of the research, results of the initial probability model, and a discussion of the challenges encountered.

3. REVIEW OF RELEVANT LITERATURE

Warranty is a complex topic that stretches beyond product reliability. In order to appreciate and understand warranty issues, it was important to give a comprehensive background of warranty from a historical perspective as well as a review of the current reliability tools. Since the majority of current reliability tools are focused on one time analysis, the purpose of this literature review will be to seek to develop the reader's knowledge of the current tools that are used today and relevant research that seeks to extend this one-time analysis into a predictive model will be discussed.

Therefore, there will be an initial discussion on the background of warranty that will cover the evolution of warranty as well as the different types of warranty policies available. Although not the focus of this research, the purpose of the background will be to introduce the reader to the legal aspects of warranty. This will be followed by a review of current reliability prediction tools covering popular reliability assessment tools (FMEA, FTA, etc) as well as physics of failure and reliability block diagrams. These are tools that are currently used to perform a one-time reliability analyses. Although these tools were not utilized in this research, the possibility of incorporating this into a Bayesian predictive model is discussed in the future work. The last review section covers relevant research, parts of which were later used in this thesis.

3.1 Background

3.1.1 Evolution of Warranty

In order to understand warranty events, it was necessary to recognize the evolution of warranty and the impact it had on building customer relationships. Warranty, in essence, is a promise that a seller makes to the buyer concerning the quality of goods or their fitness for a particular purpose. It is therefore important that a customer's expectation of a product's purpose is aligned with the intended purpose of a particular product. Depending on the context that it is used, warranty can be (Arvinder, 1998):

1. Law in a contract, a promise or binding statement which is non-essential to the main purpose of the contract, so that a failure to honor it does not cause the contract to be ended but may give the other party good reason to claim damages for breach of warranty.

2. Insurance, a statement by the insured declaring that facts given by him are true and that the insurance contract may be void if any of these facts prove to be untrue.

3. Commercial, a promise or statement by the seller or the buyer concerning the quality of goods or their fitness for a particular purpose. Without warranty, the goods are being sold on the condition that the seller has no responsibility for any faults or imperfections in the goods, and the buyer has no right to return them or claim damages or any other remedy.

Therefore, when describing warranty and warranty events it is vital that the issues of negligence, fault and/or due care are discussed and understood by all parties involved. For example, a seller or manufacturer may be liable for a defect whether he/she knew it or not but will not be if there is a breach of warranty. These issues are important when understanding the legal ramifications of developing a product's warranty.

In early civilizations, the issue of warranty was raised from a variety of products from cattle to slaves. Tablets from Babylonia have been found to have read (Arvinder, 1998):

> *…If a man has bought a male or female slave and the slave has not fulfilled his month, but the bennu disease has fallen upon him, he (the buyer) shall return the slave to the seller and the buyer shall take back the money he paid…*

This sort of "money back guarantee" from the Hammurabic Code offered the buyer compensation for defects discovered in the product after the sale. For various other products and services, the Hammurabic Code provided an eye-for-an-eye type of compensation, for example, a house builder, "who has not made strong his work" (Arvinder, 1998) causing the house to collapse thereby killing the owner, is put to death for his negligence. Codes regarding warranty events during these ancient civilizations had varying time periods from which claims may be made.

Ancient Indian law dealt with warranty events similar with that of the Babylonians, "money back guarantees" were provided to dissatisfied buyers in a specified time period. These time periods were for example: iron (one day), milking cows (three days), and beasts of burden (five days) (Arvinder, 1998). In contrast, Islamic law handled warranty events from a religious perspective, placing emphasis on intent.

Roman law, formulated under 12 tables (fundamental laws of the land) dictated that in order for a seller to trade products in the open market, he/she must disclose any and all defects and promise that no other defects existed. This provided the seller limited protection as he/she may refuse to take back a product if no defect existed on the day of the sale. Jewish law provided for an "implied" warranty, all property transactions carried: a guarantee of good title against the entire world, a warranty that seller had not encumbered the property, and a guarantee against any personal claim (Arvinder, 1998).

From the Industrial Revolution and beyond, protection for the buyer decreased with the growing acceptance of *caveat emptor* or "let the buyer beware". Under this idea, buyers were not entitled to receive compensation for any problem associated with product except outright fraud on the part of the seller. Although this may seem unfair to the buyer, in most cases the issue was moot as the buyer and seller were usually from the same local community, and there usually was no need for an express warranty. It would be far more appropriate and effective for buyers to have expressed their dissatisfaction on a personal level. It was not until the late nineteenth century that standardized product warranties became common. At the start, product warranties were almost always one-sided, providing little to no protection for the buyer and most likely did not cover failed component parts, transportation charges, ensuing damages, etc. In addition, most companies failed to honor warranties, and a trend of dishonest companies caused customers to perceive warranties as an indicator of poor product quality. It was not until the development of several independent product-testing organizations that these types of practices were curbed. These testing agencies are still around today, such as Underwriters Laboratory, Good Housekeeping Institute, and Consumer Reports. Seals of approval from these independent testing agencies went a long way to gain consumer confidence for a particular product (Arvinder, 1998).

In the United States, the Federal Trade Commission (FTC) created several laws governing the sale of goods. The Uniform Sales Act enacted during the 1930s defines warranty as:

> *… any affirmation of fact or any promise by the seller relating to the goods… if the natural tendency of such affirmation or promise is to induce the buyer to purchase the goods, and if the buyer purchases the goods relying thereon*

This definition highlights the obligations of express warranty, the two kinds of which is promissory or contractual in nature and which is the nonpromissory affirmations of fact.

It could be said that throughout the evolution of trade, product warranty has evolved and maintained a significant position in trade practices of various societies through the ages. As stated above, warranty is more than product failure per se. It is a contract, insurance, and an advertisement of the product's quality. This highlighted the need to look at warranty from a multi-faceted approach. However, while the legal aspects of warranty were relevant and interesting, the focus of this research was on the technical and user satisfaction elements of warranty.

3.1.2 Warranty Policies

Although warranty policies were not the main focus of this thesis, it was important to develop the reader's understanding of warranty by reviewing the different types and variations of warranty policies available on the market today. Blischke and Murphy (1992) gave a good representation of the types of warranty policies found, in Figure 2, shown below.

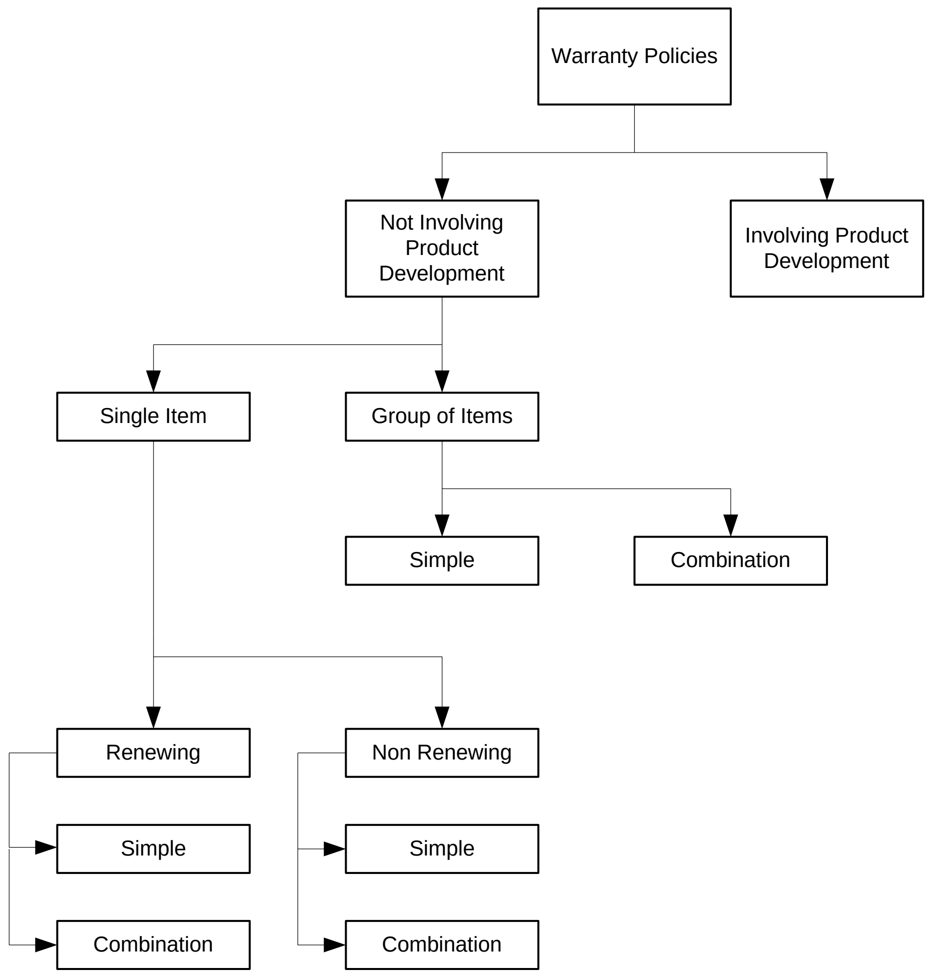

Figure 2. Warranty Policies (Blischke & Murthy, 1992)

For most consumers, the type of warranty policy that they will be familiar with is the simple non-renewing single item not involving product development warranty policy. This warranty policy is typical for items such as consumer electronics that come with a one year warranty. For these products, the product is warranted against manufacturer defects for one year. This generally means that the manufacturer will service or replace the unit if found defective for free until the one year limit. This type

of policy is generally non-renewable, if the unit was found defective 6 months into the warranty period with a new unit, the new unit does not extend the warranty by another year.

For large equipment consumers such as the military, warranty policies include a contract that involves product development after the sale. This is because, the item is generally large and complex (ex. Aircraft carrier) and it is understood that the item will most likely need to be modified or improved after sale rather than be replaced.

Although the appropriate design of warranty policies can help reduce the cost of a company's warranty costs, this was not the focus of this research. The focus of this research was the development of a predictive reliability assessment tool that would extend the one-time reliability analyses.

3.2 Current Reliability Prediction Tools

There are several popular reliability assessment tools that are currently in use today at many companies. Among them are Failure Modes and Effects Analysis, Fault Tree Analysis, Physics of Failures, and Reliability Block Diagram. These tools are effective at analyzing reliability at a given period of time. However, because they are traditionally performed as a one-time analysis, they are not as effective in predicting warranty or reliability performance early on during the product life cycle. This research sought to address this gap by attempting to integrate these various tools into a Bayesian framework to better predict warranty performance early in the product development. Although this was found not possible due to the information provided by our industrial partner, these tools are still good candidates as quantitative and qualitative data for the Bayesian input model.

3.2.1 Failure Modes and Effects Analysis

Failure modes and effects analysis (FMEA), a procedure for analyzing potential failure modes was first introduced by the US Armed Forces in the late 1940s. It is widely used during both product and process development to identify and analyze failure modes and the severity of their consequences. Failure modes are any errors or defects in a process, design, or item, especially those that affect the customer, and can be potential or actual. The benefit of conducting a FMEA during product development includes the ability to perform a step by step breakdown of potential failure modes and rank them by risk.

As one of the most widely used techniques in product risk analysis, it allows potential product problems to be identified before they reach the customer (André, et. al, 2008). As shown in Table 3, each component or failure mode is examined for the following: probability of failure occurrence (Occurrence), severity of failure (Severity), and ability to detect failure before it occurs (Detection). Multiplying these generates the risk priority number (RPN) which allows teams to prioritize the failure modes. If the RPN value is higher than a predetermined limit, actions (ex. design mitigations) are generally required to mitigate the failure risk. FMEAs are effective not only for their ability to prioritize critical failures but also because they require an analysis of each component of a system (André, et. al, 2008). Nevertheless, there are several drawbacks to FMEAs, as outlined below (Javier, et. al, 2002):

- Risk evaluation using RPN cannot always be assessed by “detection”;
- There is no exact rule to determine the probability of occurrence and detection;
- Calculation of the RPN based on the three measures may also be distorted. While the probability of non detection and its respective scores follow a linear function, the

relationship between the probability of failure occurrence and its score is not necessary linear;

- Different scores for occurrence and detection can result in the same RPN, despite the risks involved being completely different; and
- The RPN is not an effective measure of proposals for improvements.

These drawbacks make the FMEA a useful tool for qualitative data but not so much for quantitative. Although this research did try to leverage the qualitative data provided by the FMEA and supplement the quantitative with other sources, i.e., historical data, we were unable to do so. In the future, it may be possible to integrate the FMEA with other data sources into a Bayesian framework for warranty prediction.

Table 3. Failure Modes and Effects Analysis Example

Item/ Function	Potential Failure Mode	Failure Mode Effects	Severity	Failure Causes	Occurrence	Current Controls	Detection	RPN	Actions Taken	Severity	Occurence	Detection	RPN

3.2.2 Fault Tree Analysis

The fault tree analysis (FTA) is a failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events. Used mostly in the safety engineering field to quantitatively determine the probability of a safety hazard, it is useful in breaking down complex systems into simpler contributing components.

This top down, event-oriented approach allows the identification of basic events that cause system failure (Bailey, et. al, 2008). As shown below in Figure 4, a graphical tree structure is used to represent all events. The root of the tree is called the top event and the leaves of the tree are called basic events. Logic symbols that combine the events between the top event and the leaves allow for both quantitative and qualitative analysis.

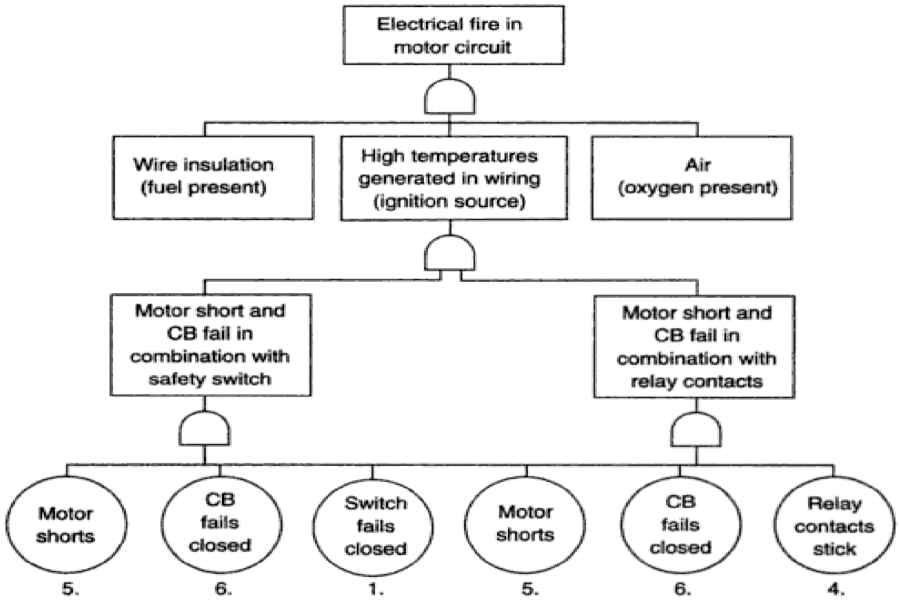

Figure 4. Fault Tree Analysis Example (Stamatis, 2003)

Although useful for taking all types of failures into account, due to its inherent structure, only a single event can be a top event. Therefore, additional fault trees must be developed for every top event. This can be a time consuming process that generally leads to only a handful of fault trees created. In addition, FTAs are usually performed once during the product development process and therefore does not accurately reflect the product as it is being developed.

Furthermore, because the value and accuracy of the FTA depends on the skill and experience of the analyst, the quality of the fault tree can vary greatly (Bailey, et al., 2008). Therefore, the FTA is a good candidate for use as part of the engineering judgment data for the Bayesian framework. Unfortunately this research was unable to find any relevant fault trees and therefore was unable to utilize this type of data.

3.2.3 Physics of Failures

The Physics of Failure approach can be used in reliability engineering by providing the “when” and the “why” for a particular failure mode. It does this by utilizing an understanding of the failure mechanisms involved, such as crack propagation or chemical corrosion. A generic P-o-F approach is shown below in Figure 5 (Matic & Sruk, 2008):

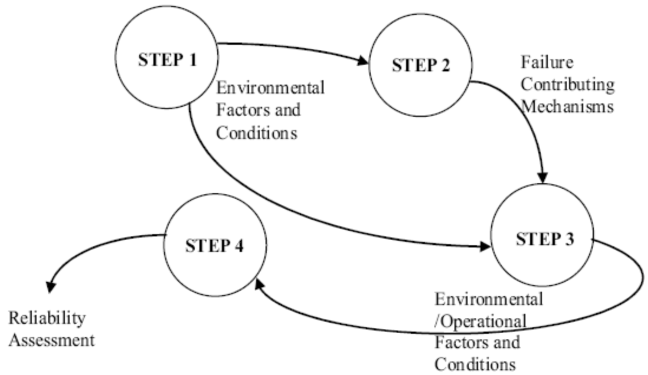

Figure 5. Physics of Failure Model (Matic & Sruk, 2008)

As illustrated above, the first step is by evaluating the environmental factors and conditions. The next step involves isolating potential failure triads (site, mode, mechanism)(Matic & Sruk, 2008) and by determining what the failure contributing mechanisms are. This step combines the identification of failure sites and site corresponding failure modes, and the determination of mechanisms contributing to a potential failure mode (Matic & Sruk, 2008). The next step filters contributing environmental and/or operational factors and the last step finds the functional dependencies of all stresses and identifies any applicable models. From the identified models, a particular one is selected that is the best fit for the specific operational/environmental conditions. Therefore, when the proper equations are known, the effect of the operational/environmental condition is also known as well.

There are many advantages to the P-o-F approach for evaluating and identifying reliability concerns and in effect positively impact the development cycle and reduce project costs. Among them are the ability to compare up-front design candidates, identify up-front design improvements, obtain realistic predictions, estimate the reliability quickly, determine the life expectancy of components, optimize environmental stress screening, and identify a focus of preventive maintenance and its optimal preventive interval.

Nevertheless, the disadvantages of the P-o-F approach include the need for detailed component manufacturing information (material, process, design data, etc), complex analysis, and the difficulty in assessing the entire system. Therefore, a P-o-F approach is unable to give a complete picture of the warranty performance of a product, especially during the early stages of the development cycle. For this thesis, the feasibility of integrating the P-o-F approach with other current reliability tools was explored. However, the industry partner selected did not have any P-o-F available for review and therefore this tool was not utilized.

3.2.4 Reliability Block Diagram

Reliability Block Diagrams (RBD) are used to perform reliability studies and provide information on system risks by evaluating the functional relationship between components in a system. These models also infer predictions based on parts-count failure rates taken from historical data. It should be understood that these predictions are rarely accurate but are useful in understanding the relative severity of risks involved. The figure below (Figure 6) shows a simple reliability block diagram (Gough, et. al, 1990):

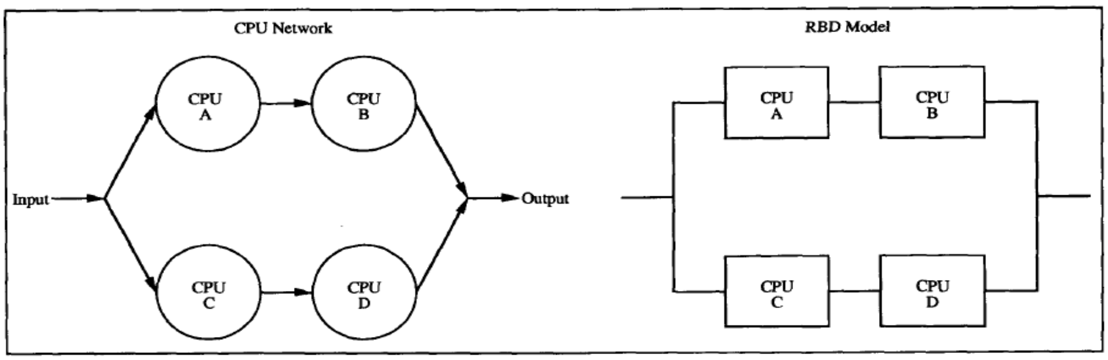

Figure 6. Reliability Block Diagram Example (Gough et. al, 1990)

RBDs are able to handle complex systems as they are easily scalable from small and simple to large and complex systems. This makes RBDs well suited to today's systems as they grow in size and complexity. In contrast to fault tree models that model component failures as they relates to system failure, RBDs focus on how component success results in system operation success.

Like a FTA, RBD provides an intuitive graphical representation of the system from a reliability perspective (Bailey, et al., 2008). However, for all their benefits, RBDs are only as accurate as the failure data available for the components that make up the system (Bailey, et al., 2008). Failure rates that assigned to a component in the system may not accurately reflect what is actually occurring in the field. In addition, RBDs also assume that items fail independently from each other which may not be the case.

For this thesis, the possibility of using RBDs was explored as a way to provide engineering judgment data, along with FMEAs, FTAs, and Physics of Failure approaches. By integrating data from the RBD with other current reliability tools, a more complete picture of future warranty performance can be predicted. However, although it would have been beneficial to combine an RBD approach with an FMEA to link multiple causes to a single failure event, the industry partner we selected did not have this data available and therefore we were unable to utilize RBDs.

3.3 Related Research

3.3.1 Framework for Reliability Prediction

Yadav, et al. (2002) described a process for reliability assessment and prediction during the product development process which could utilize qualitative (fuzzy) information, prior knowledge, and quantitative data. Ideally by integrating all existing reliability assessment data, we could achieve better accuracy and realistic estimates. In order to effectively track and manage reliability improvement during the development phase, continuous reliability estimation is necessary as product moves from one design phase to another. Yadav, et al. (2002) incorporated the fundamental Bayes' theorem with fuzzy logic reasoning to enhance the capability of the Bayesian model to accept fuzzy information. This was due to the subjective and qualitative nature of engineering judgment, as well as other factors that did not provide hard numerical data.

Mazzuchi and Soyer (1993) described testing performed at each test stage as a basis for defining reliability growth. Yadav, et al. (2002) presented the idea of viewing each test stage as an input-output model shown below in Figure 7. If possible, reliability growth data suggested by Mazzuchi and Soyer (1993) would be integrated at each development stage with all other available data such as engineering judgment (fuzzy input), prior distribution, and quantitative data. The outputs at each stage would be the reliability estimates and the posterior distribution. This posterior distribution could subsequently be used at the next stage as the prior distribution. Although Mazzuchi and Soyer's (1993) approach accurately captured reliability through product development, it did not address warranty issues that may not be related to component reliability, such as mismatched customer expectations. This is a gap in the current literature that this research sought to address.

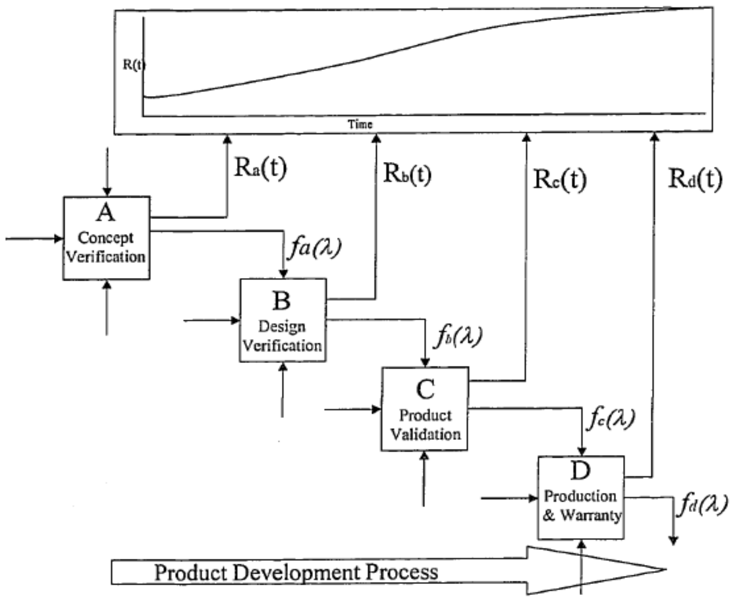

Figure 7 Product Development Process Model with Reliability Growth (Yadav & Prakash, 2002)

Yadav, et al. (2002) suggested that by calculating the reliability estimate at each design phase, it would be possible to increase product reliability over time. This would result in a revised reliability estimate at the end of each stage incorporating the engineering judgment for design changes, corrective actions, and other qualitative information. Ideally, this estimate would show a positive change in reliability improvement at the end of each stage. Although Yadav, et al.'s (2002) work did a good job incorporating qualitative (fuzzy) information, prior knowledge, and quantitative data, it did not take into account the warranty scenario as a chain of events. Yadav' focus was on the product reliability only and did not take into account warranty events out in the field.

Esterman, et al. (2005) expanded on the work discussed by Yadav, et al. (2002), and suggested a new framework that consisted of the following components (shown below in Figure 8): warranty scenario identification, warranty event generation, warranty scenario costs, and prioritization & risk mitigation. Similar to the model that Yadav, et al.'s proposed, the event generation engine would be able to generate the probability of the warranty event from the identification of the warranty scenario using data from multiple sources.

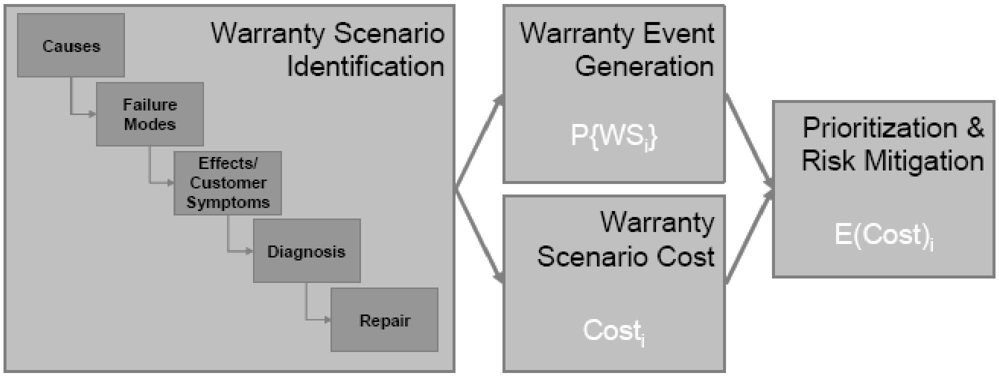

Figure 8. Framework for Predicting Warranty Performance (Esterman, et al., 2005)

This type of framework would allow warranty performance predictions during the product development phase by accomplishing the following objectives (Esterman, et al., 2005):

1. Facilitate decision-making by increasing the product developers' and managers' confidence that their actions are leading to improved warranty performance in the field. In addition, these models should provide insights to the development team for actions they can take to mitigate warranty costs.

2. Provide the management team an accurate projection of warranty costs so that the enterprise can appropriately plan for the financial impact of these costs. These impacts include: product pricing, extended warranty support pricing, service inventory requirements, warranty accruals, etc.

This research sought to further develop the framework proposed by Esterman, et al. (2005) by researching the feasibility of combining multiple information sources into a Bayesian process. Data mining techniques and Bayesian methods was also used to be implement this framework on a test case study to demonstrate feasibility.

3.3.2 Bayesian Statistics

Bayesian statistics is a statistical theory and approach to data analysis that provides a coherent method for learning from evidence as it accumulates(Campbell, 2006). Utilizing Bayesian statistics, we can develop a systematic framework that can accommodate noise, variability, and low samples sizes. This will allow us to integrate disparate incomplete datasets throughout the product development cycle.

Therefore, a review of the general mechanics and applications of Bayesian statistics was important as they provided insight into how Bayesian statistics could be used to create a model that would

predict warranty events early in the product development cycle. Furthermore, an understanding of the usage of Bayesian statistics could reveal data conditioning requirements for integrating prior historical knowledge, engineering judgment, and test results.

The fundamental idea in Bayesian statistics is that one's uncertainty about an unknown quantity of interest is represented by probabilities for possible values of that quantity(Campbell, 2006). Using the Bayes theorem (shown below in Figure 9), it would be possible to combine previous information with current data.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}.$$

Figure 9. Bayes' Theorem

This is done by defining the probabilities of the Bayes Theorem. For instance, before testing begins and data is obtained, a prior distribution P(A) can be determined based on prior probabilities or a distribution that matches the failure mode. For example, P(A) can be the probabilities of a particular event according to prior historical data. If there is no previous knowledge, a non-informative prior distribution can be used instead. As data is gathered and information is collected, prior probabilities P(A) are updated per Bayes' Theorem to posterior probabilities P(A|B). These posterior probabilities P(A|B) are probabilities for values of the unknown quantity after data is observed and in the next iteration take the place of the prior probability.

In doing this, a Bayesian approach allows for the derivation of the predictive probability from a posterior probability. This predictive probability is the probability of future events given outcomes that have already been observed. With all possible values of future outcomes, we can create a predictive distribution that will permit us to determine when to stop testing, predict outcomes, and adjust test results for missing data. Bayesian statistics is different than traditional frequentist approaches in that Bayesian analysis bases all inferences on the posterior distribution which is the product of the prior distribution and likelihood function.

In the past decade, Bayesian methods have become more commonly used than ever before (Campbell, 2006). Due to their ability to update probabilities as data is accumulated, this research used Bayesian statistics as the basis for the warranty prediction model.

3.3.3 Bayes During Development Testing

Although not focused on warranty per se, Mazzuchi & Soyer (1993) presented a method for analyzing product reliability during the development phase using a Bayes approach. During product development, testing is performed at various stages to determine if a design change is needed. These design changes are made in hopes of enhanced product durability and/or quality. This type of process generates both attribute (pass/fail) and variable (failure time) test data that can be termed “reliability growth”. This reliability growth is the building block for the model developed by Mazzuchi & Soyer (1993) to determine a product’s reliability at each stage of the design.

In the proposed Bayes approach, failure data from sequenced testing/modification stages is used as prior distribution. Test results at each test stage can be used to update this probability and therefore affect the probability at subsequent test stages leading up to the final development phase. This is an important distinction between other current Bayesian approaches to product reliability as it yields future reliability estimates after each test stage (Mazzuchi & Soyer, 1993) instead of the current reliability estimate at each test stage.

Mazzuchi & Soyer (1993) suggested a framework for incorporating prior information using the prior distribution shown below where *m* specifies test stages, *q* is the probability of occurrence for a nonfixable-cause failure mode and *p* specifies the testing performed. This type of framework allowed for actual test-stage failure probabilities and perceived absolute and/or relative change in these values (Mazzuchi & Soyer, 1993) shown below in Equation 1.

$$\pi(q, p \mid \beta, \alpha) = \frac{\Gamma(\beta)}{\prod_{j=0}^{m+2}\Gamma(\beta\alpha_j)}\left\{\prod_{j=0}^{m+1}(p_j - p_{j+1})^{\beta\alpha_j - 1}\right\}$$

$$\cdot(1 - q - p_1)^{\beta\alpha_0 - 1} q^{\beta\alpha_{m+2} - 1}$$

where

$\mathrm{m} = \text{number of stages}$ (1)

$\mathrm{q} = \text{nonfixable} - \text{cause failure in a p} - \text{test}$

$\mathrm{p} = \text{fixable - cause failure in a p - test at test - stage}$

$\alpha_i, \beta = \text{prior positive parameters}; \sum_i \alpha_i = 1$

This type of framework was different from other authors who advocate selection of the prior parameters based on queries about observable quantities such as the number of successes, nonfixable-cause failures, and fixable-cause failures in a test stage(Mazzuchi & Soyer, 1993). The focus here was on the perceived reliability growth pattern as a whole.

While Mazzuchi & Soyer's research is similar to this research in that both use the Bayes theorem to integrate failure data at various points during the product development, Mazzuchi & Soyer's research focus primarily on product failures. Mazzuchi & Soyer also focuses primarily on failure data and did not present a way to integrate disparate data sets that could also add value to the development of the posterior probability. This research sought to address these two opportunities.

3.3.4 Bayesian Belief Networks

A review of Bayesian Belief Networks was important as they helped us understand how warranty events could have causal relationships that are determined by both qualitative and quantitative data. Although not used in this research, they present a way to assign probabilities to a chain of events that could include the warranty event and repair.

Bayesian belief networks (BBN's) are also known as "probabilistic networks" and constitute a mathematically sound way for representing and reasoning with joint probability distributions (JPDs) in an internally consistent manner (Lee, 2002). Many top companies have successfully utilized BBNs to diagnostically model mechatronic equipment. A simple Bayesian belief network is shown below in Figure 10 (Lee, 2002):

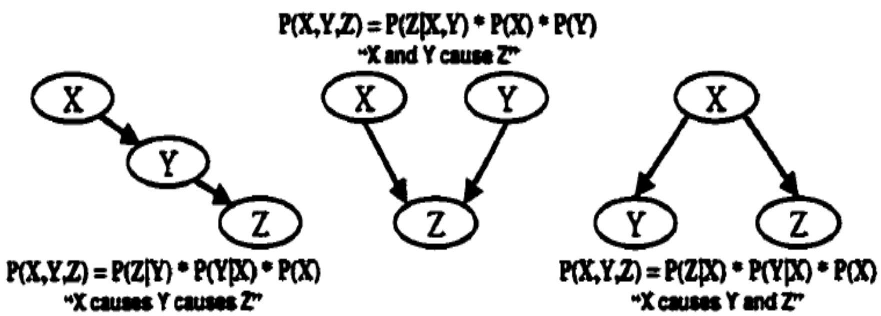

Figure 10. Simple Bayesian Belief Network Example (Lee, 2002)

A Bayesian belief network consists of the following qualitative and quantitative relationships: set of random or deterministic variables, set of directed edges or arcs, and a set of root and conditional probabilities. From the example above, there are variables (Sprinkler, Rain, Grass Wet), the causal relationships between them (shown by the arrows), and the probabilities describing it all. There are three fundamental "causal patterns" that are the basis for how all other patterns are constructed:

- X causes Y causes Z
- X and Y cause Z

- X causes Y and Z

Exacting inferences from BBNs may be performed with either exact or approximate methods depending on the structure used (Lee, 2002). An exact method such as the Probability Propagation in Clique Trees (PPCT) method works in two basic steps. The belief network is first converted into a secondary computation structured called a "clique tree" and then the probabilities of interest are computed by operating on that secondary structure. BBNs present an opportunity to model multiple causes to a single failure mode. This is an advantage over most current reliability tools such as the Fault Tree Analysis. By relating probabilities to failure events, it may be possible to generate likelihoods for individual warranty events.

3.3.5 Failure Modes and Effects Analysis with Bayesian Belief Networks

Although not used in this research, Burton Lee's research presented a way to incorporate Bayesian techniques into a popular reliability analysis tool, Failure Modes and Effect Analysis, in order to better quantify risk. While this approach has its merits, it fails to address other sources of data such as life test data or engineering judgment. Nevertheless, it was important to review relevant literature in order to discover parallels between this research and Burton's.

A common criticism of scenario based FMEAs from experienced engineers is the lack of correlation between the criticality rating and real life scenarios. Although RPN values are helpful in prioritizing risk, it appears to do a poor job accurately quantifying it. Combining FMEAs and Bayesian Belief Networks may be one way to rectify this.

The BN-FMEA uses the belief network theory to construct directed acyclic graph (DAG) models of failures scenarios to represent causal and statistical dependences between internal and external states as well as the event variables of the physical system (Lee, 2002). It uses a new class of severity variables as well as both root probabilities and conditional probabilities to obtain improved inference and design trade-off evaluation as compared to a traditional FMEA. Similar to a traditional FMEA, a FMEA criticality matrix is generated from the belief net model.

A Bayesian network based FMEA methodology allows for the specification of severity distributions, based either on a relative or absolute severity standard. Severity distributions provide more information about failure scenario impacts and their potential prioritizations than do criticality matrices.

The BN-FMEA method utilizes the Bayesian network graph with four primary variable groupings: physical system, customer and world state and event variables, and severity variables. In order to construct this model, Burt Lee suggests the following steps (Lee, 2002):

> Step 1 - Build failure scenarios: Failure "chains" representing individual failure scenarios are constructed out of these physical system variable types. System-level models are assembled from chains which share variables; (Lee, 2002)
>
> Step 2 - Severity annotations: For every variable designed as a "failure end-event" or FEE, attach a severity variable along with its associated parent variables; (Lee, 2002)
>
> Step 3 - Compile the Clique Tree: compile every failure scenario-severity model; (Lee, 2002)
>
> Step 4 - Extract and Plot: Extract the required failure occurrence probability and severity distribution information from each compiled scenario and plot on the criticality matrix. (Lee, 2002)
>
> Step 5 - Update: revise the failure scenario model to reflect any design improvements made as a result of the FMEA analysis. (Lee, 2002)

In order to build a failure scenario, we need to start with the "failure chain", the complete failure sequence of the physical system, from the original cause to intermediate effects, and finally to the "end effect" of "failure end-event". There are two basic approaches to developing a failure scenario, the component-based variable identification and the function-based variable identification.

Conditional Severity Variable

Severity Distribution = P(Severity|failure end-event, system-internal states, system-external factors, customer actions or states)

The benefit of utilizing a probability distribution for severity lends itself to flexibility. Traditional FMEAs define severity as a point distribution which by definition lends itself to a potentially wide range of interior and exterior states and events.

Dual failure scenarios and Severity

When more than one failure event occurs, the severity variable is the total severity realized. We are able to do this as all severities in a given model are of the same form with the same unit of measure. We can also expand this to include multiple failure scenarios.

We can set the use of a single severity standard across all failure scenarios that must be enforced by the modeling environment during model construction.

Generation of the BN-FMEA Criticality Matrix

The Criticality Matrix as well as the candidate pointset (P_i) is generated once all qualitative and quantitative relationships are specified in their respective DAG's for all failure scenarios. For each failure scenario, a single candidate point Pi is assigned to it. In order to determine the *x* and *y* coordinates for each candidate point, the y-axis is set to the failure end-event's a priori value from the clique tree. By setting the failure end-event (FEE_i) to what is observed and obtaining the maximum severity state, the x-axis is also obtained.

Registering Design Improvements

Individual failure scenarios are updated as design improvements are identified during the FMEA process. As expected, reliability improvements will result in the decreasing of the underlying root probabilities or conditional probability tables. Major design changes may result in changes to the DAG, for example new nodes or modified ones.

3.3.6 Data Mining

Data mining, also known as the knowledge discovery from data (KDD), can be defined as the application of computer algorithms to discover useful knowledge in large databases (Romanowski, 2004). Although it is not a new field, having been in use for over ten years, it is a tool that is heavily used in a variety of industries, such as the banking industry. A data mining approach can be symbolic or non-symbolic; predictive or classifying; but it is always interactive and iterative (Romanowski, 2004). Regardless of the particular approach used, the following general steps are the same for most data mining algorithms (Romanowski, 2004):

- Determine the type of learning
- Choose the data mining algorithm
- Choose the target variable
- Pre-process the data
- Mine the data
- Analyze the output
- Refine the task

As data sets have grown larger and larger necessitating the use of automated computer systems, data mining algorithms have become important in extracting useful information. Although there are numerous data mining algorithms available, several popular ones will be discussed here.

3.3.6.1 Clustering

The k-means algorithm, also called k-means clustering is a technique of grouping a number of n observations into k clusters depending on how close an observation is to the cluster with the nearest mean (otherwise known as seeds). It is an iterative method that allows the user to at first select a random k initial means, then associating every observation with the nearest mean. The centroid of each of the k clusters (seed) becomes the new means and existing k clusters are redefined repeating until convergence has been reached. See Figure 11 below (Berry, et al., 2004):

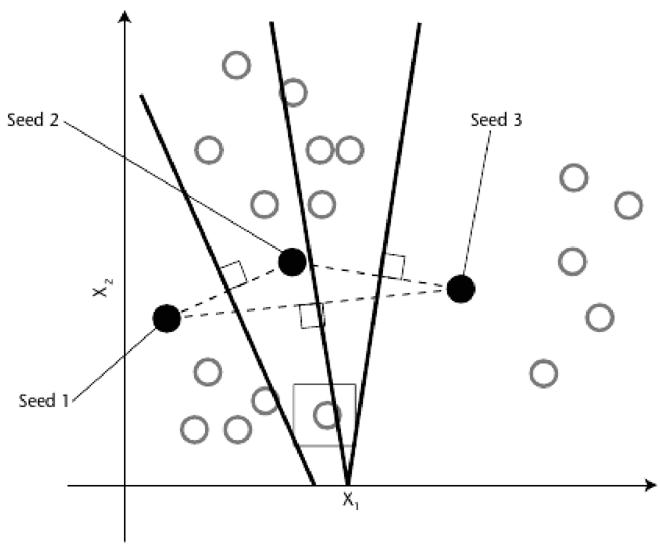

Figure 11. K-Means Algorithm Example (Berry, et al., 2004)

3.3.6.2 Support Vector Machines

Support vector machines (SVM) are another data mining algorithm that utilizes classifiers to divide a group based on the concept of decision planes that define decision boundaries. SVM constructs a linear model to estimate the decision function using non-linear class boundaries based on support vectors. If the data are linearly separated, SVM trains linear machines for an optimal hyperplane that separates the data without error and into the maximum distance between the hyperplane and the closest training points. The training points that are closest to the optimal separating hyperplane are called support vectors (Kim & Sohn, 2008). Considered robust and accurate, methods for training SVM are being developed at a fast rate. The image shown below in Figure 12 is a classic example of a linear model where the classifier separates a set of objects into their respective groups (Blue and Grey) with a line. As most classification tasks are not as simple as the example shown, more complex structures are needed in order to make an optimal separation and also correctly classify any new objects (test cases). SVM need examples (train cases) in order to work properly on accurately classifying new objects. Although there are benefits to

using SVMs, disadvantages include: slow speed during the test phase, selection of the kernel function parameters, high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks (József & Gábor, 2008).

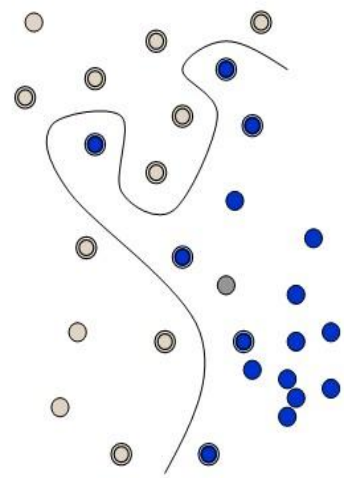

Figure 12. Support Vector Machines Example

3.3.6.3 Association Learning

The Apriori algorithm is a popular data mining approach that learns association rules. It attempts to find similar items given a set of itemsets. It does this by finding subsets which are common to at least a minimum predetermined number of the itemset. Apriori algorithms are typically used for transaction based datasets, such as determining what items a customer buys and if there is a relationship between the items customers buy, such as a computer and a keyboard. Depending on the type of values, the association rules can be classified into either Boolean Association Rules or Quantitative Association Rules. Regardless of the association rule selected, the following determines the association (Wu, et al., 2008):

- Minimum Support Threshold
    - The support of an association pattern is the percentage of task-relevant data transactions for which the pattern is true.
- Minimum Confidence Threshold
    - Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern.

Apriori first scans the database and searches for frequent itemsets of size 1 by accumulating the count for each item and collecting those that satisfy the minimum support requirement. It then iterates on the following three steps and extracts all the frequent itemsets (Wu, et al., 2008). Given a historical data set, an Apriori algorithm may be one possible way to identify relationships between features. Other types of data mining algorithms are tabulated in Figure 13.

Algorithm	Learning Type	Classification	Estimation	Association	Memory Requirements	Training Time (CPU seconds)	Testing Time (CPU seconds)	Ease of interpretation	Batch only	Type of Data	Accuracy	Scalability
Linear regression	S	N	Y	N	Very low	Fast	Very fast	Easy	Y	Numerical	Medium	High
Neural nets	S	N	Y	N	Low-med	Very slow	Very fast	Very hard	N	Numerical	High	Low
Rule induction	S	Y	N	N	High	Fast	Very fast	Easy	Y	N/C	Med Low	High
Decision trees	S	Y	Y	N	Low	Very fast	Very fast	Very easy	Y	N/C	Med Low	High
Case-based reasoning	S	Y	Y	N	High	Very fast	Fast	Easy	N	N/C	?	High
Kth nearest neighbor	S	Y	Y	Y	Very high	Very fast	Slow	Easy	N	N/C	MedLow	Very Low
Self-organizing maps	U	N	N	Y	Low-med	Fast	NA	Hard	Y	N/C	NA	Low
Bayesian belief nets	U	Y	N	Y	?	Slow	NA	Hard	Y	N/C	?	?
Support vectors	S	Y	Y	N	Medium	Slow	Medium	Easy	Y	Numerical	High	?

S=supervised; U=unsupervised; Y=Yes; N=No; NA=not available; N/C=numerical and categorical; ?=unknown;

Figure 13. Typical Data Mining Algorithms and their characteristics (Romanowski, 2004)

3.3.6.4 Decision Trees

Decisions trees are one way to model a group of conditions and their result in a tree-like format. With a decision tree, it is possible to predict a result even if the set of conditions is not in the original dataset.

Given a decision table (Figure 14), we can use a “divide-and-conquer” approach to the problem of learning from a set of instances which would lead us to develop a decision tree (Figure 15). Nodes (represented by a diamond in Figure 15) in a decision tree involve testing a particular attribute, normally comparing an attribute value with a constant (Witten & Frank, 2005). The leaf nodes (represented by a square in Figure 15) give a classification that applies to all instances that reach the leaf node, or a set of classifications. To classify an unknown instance (new data), the data follows the tree according to the values of the attributes tested in successive nodes, and thus when a leaf is reached, the instance is classified according to the class previously assigned to that leaf (Witten & Frank, 2005).

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Figure 14. Weather Example Dataset (Witten & Frank, 2005)

In building a decision tree, there are two approaches: top-down construction and bottom-up tree pruning. In a top down tree construction, all training examples are at the root and examples are partitioned recursively by choosing one attribute each time. In a bottom-up tree pruning, branches or sub trees are removed in a bottom-up manner to avoid over fitting. To choose the splitting attribute, all available attributes are evaluated at each node on the basis of separating the classes of training examples. One of two good functions are typically used for this purpose: information gain and information gain ratio. A good criterion for attribute selection is the one that will result in the smallest tree. This will reduce the risk of overfit, where the tree has defined the data set too strictly. It may however result a risk of underfit, which results in the opposite problem. Using a goodness function such as information gain, we can select attributes based on the average purity of the subsets that the attribute produces.

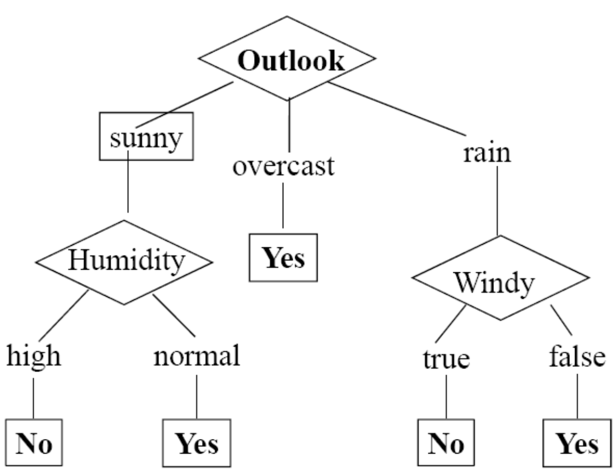

Figure 15. Decision Tree for Weather Example Data (Witten & Frank, 2005)

3.3.6.5 Data Mining Conclusions

While data mining can be used to uncover patterns in data samples, it is important to be aware that the use of non-representative samples of data may produce results that are not indicative of the domain. Similarly, data mining will not find patterns that may be present in the domain, if those patterns are not present in the sample being "mined". Hence, an important part of the process is the verification and validation of patterns on other samples of data.

3.3.6.6 Data Conditioning

A major focus of this research was the integration of different datasets for input into the Bayesian model. In order to do this, a common set of parameters was needed. After this was set, data could then be converted or transformed to meet these criteria.

3.3.7 Design for Warranty Cost Reduction

During product development, a process based cost model for warranty events can be used to reduce the eventual cost of warranty. This can be successfully done when key warranty cost drivers are identified and a set of cost reduction strategies are executed.

With the total cost of warranty for computer and related high technology US based companies now approaching $8 billion per year (Mueller, 2007), design for warranty cost reduction takes an important in role in maintaining profit margins.

Although numerous companies have shifted their warranty costs by moving towards an extended warranty strategy, this does not address the root causes and effects of poor product quality or misaligned customer expectations.

In order to reduce warranty costs, one must realize that for every warranty event there is not only a possible component replacement cost but a service process cost involved. In recent years, the available options of service processes include but are not limited to the following: phone support, web-based and customer self-fix schemes, repair centers, and on-site service calls. These service processes do not come cheap; from $30/call for a warranty event resolved over the phone to >$700 for on-site repairs (Mueller, 2007).

Therefore, development teams need to be able to design products that are both less costly to repair and more reliable.

Service Process Based Warranty Cost Model

A service process based warranty costs model addresses both the customer's problem and the support process used to resolve it. For every warranty event there is a unique linkage of a diagnosed problem, a specific support process to resolve the event and specific material costs (components), if consumed. Therefore, the total warranty cost for a specific warranty type is the following (Mueller, 2007):

Expected warranty cost = F_i * (process cost + material cost)

where F_i = frequency of occurrence of a specific warranty event type.

However, as experience shows that for most products, only a handful of warranty events dominate the total warranty costs, we can use the Pareto principle to define M, where M is the number of warranty event types that account for an acceptable percentage, for example 90% of a product's total warranty costs. Although the service process based warranty cost model was not utilized in this research, this research reviewed various costs that contributed to high warranty costs providing the basis for future research.

3.4 Literature Review Summary

In order to develop methods to condition available data streams for use in a Bayesian framework, a thorough literature review was needed to explore the vast number of tools available today. This thesis may not have used all the tools covered in this literature review, but it was important to review the applicability of each tool for this approach for their usefulness in providing good data sources. Integrating multiple data streams and data mining techniques have been explored for this thesis to identify patterns and relationships between attributes that may provide valuable information to product development teams. This would hopefully lead to data conditioning requirements for a successful warranty prediction early on in the product development process.

In summary, the literature review revealed many tools that could be used as input in the research. However, a review of current literature showed that although current reliability tools may help reduce component or product failure modes, these tools do not address other possible warranty events that may occur such as misaligned customer expectations. This is one area that can still be explored using warranty scenarios and a Bayesian approach to probability models.

4. RESEARCH METHODOLOGY

This thesis sought to develop a mathematical framework to integrate Bayesian methods and data mining techniques to develop the event generation engine in an engineer to order environment. The purpose of this research was to show feasibility that disparate data sources can be conditioned for input into a Bayesian model to predict event rates. This built on the work developed by Yadav & Prakash (2002). This research relied on the use of actual data gathered with the help of an industrial partner. Actual data allowed us to uncover additional issues that needed to be solved and therefore helped to lead to a more robust event generation solution. The steps used to execute this research plan are summarized in Figure 16.

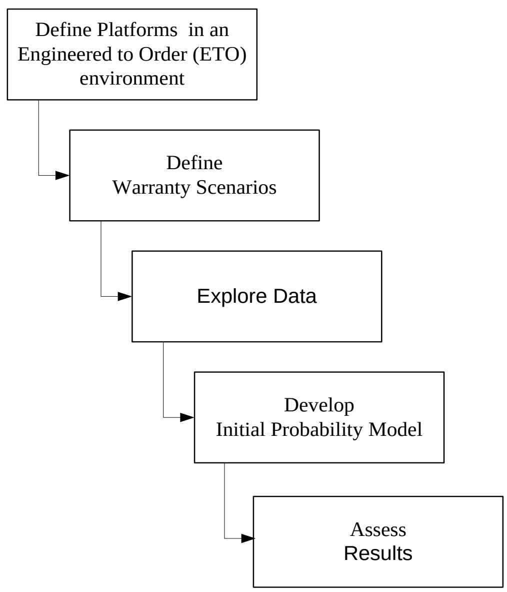

Figure 4. Pictorial Representation of Research Steps

This research started with the development of platforms in an engineered to order environment (Figure 16). In an ETO environment, each product was essentially unique. Platforms allowed us a way to aggregate data for each group of products. The development of platforms placed similar products into groups. The conditions for doing this were determined with input from an industrial partner. From this, warranty scenarios could be developed for each platform or product group based on interviews with engineers and various company personnel. Each warranty scenario described the chain of events from cause to effects to repair and diagnosis. Data was then explored for suitability as inputs to the model. After that, the initial probability model was developed using both the data explored and the warranty scenarios that were created. The results of the initial probability model were analyzed, followed by a

discussion of the challenges that were encountered. The remainder of this section describes these steps in more detail.

4.1 Platform Concept in an Engineer-to-Order (ETO) Environment

In an engineered-to-order environment, companies try to satisfy custom needs with custom products (Jin & Thomson, 2003). The major characteristics of an ETO environment are: customer involvement in product design and/or configuration, manufacturing planning directly linked to the details of customer orders, and material ordering and production scheduling driven by the pace of engineering development. These characteristics make manufacturing planning (orders, materials, facilities, personnel) complex due to the use of product information which is largely unknown at the acceptance of an order and which continuously changes as product specifications are finalized. This is in contrast to make-to-stock firms that produce market based products in very large quantities where the design characteristics are based on the market and not on an individual customer.

Rahim et al. performed a comprehensive study on the product design characteristics associated with ETO firms. Those features that are relevant to the warranty prediction process are highlighted in Table 17. A couple characteristics of the ETO system that increase the likelihood of warranty issues are increased system complexity and a lengthened product life-cycle. However, from an organizational perspective, a dependency on a similar product and a high involvement of manufacturing in design would lower the likelihood of warranty issues. These characteristics were important issues to consider because it indicated how different and important warranty issues would be to an ETO firm compared to a market-based firm.

Criteria	Characteristic
Design	Usually Exclusive To 1 Customer
Frequency Of Design	Very Frequent
Design Effort & Cost Per Product	High
Chance Of Design Improvement During Manufacture	Low
Involvement Of Manufacturing Engineers In Design	Always
Design Dependency On Similar Product	High
Customer Input During Design	Usually High
Customer Approves Design	Yes

Product Test & Commissioning	Usually At Customer Site
Customer’s Technical Knowledge	High
Certainty Of Customer Requirements	High
Product Complexity	Generally High
Customer Requirements	Generally Technical & Specific
Interpretation Of Customer Requirements	Direct
Product Life-Cycle	Long

Table 17. ETO Design Characteristics

Understanding the different design characteristics of ETO firms gave us an idea of what factors could increase or decrease the likelihood of warranty issues. But we still needed a way to group similar products together. This was because in an ETO environment, companies make a mixture of completely new products and reconfigurations of existing designs. Also, after a contract is awarded, there is a continuous cycle of design, material change/confirmation and shop floor schedule change/confirmation as designs are negotiated with the customer and completed (Jin & Thomson, 2003). These products are assigned a unique contract number which can run in the thousands depending on the volume of the manufacturer. With so many unique part numbers, there was a need to group them into a manageable number.

Product family and product platform design is a way to facilitate mass customization by redesigning and consolidating a group of distinct products based on a set of common features, components, and subassemblies (Simpson, 2004). Over time, data for each iteration of the platform (ex. Rev A, Rev B, etc) can be captured to make references for future iterations (Figure 18). This allows us to take data from one platform iteration (ex. Product Family 1, Rev A) and use that data to predict the probability of a warranty event for a future platform iteration (ex. Product Family 1, Rev B). This is important as while a product is under development (ex. Product Family 1, Rev B), it would be useful to obtain the data from a previous iteration (ex. Product Family 1, Rev A) to make inferences of future warranty events. From a model standpoint, each iteration can be considered as a variation of the variable “n”.

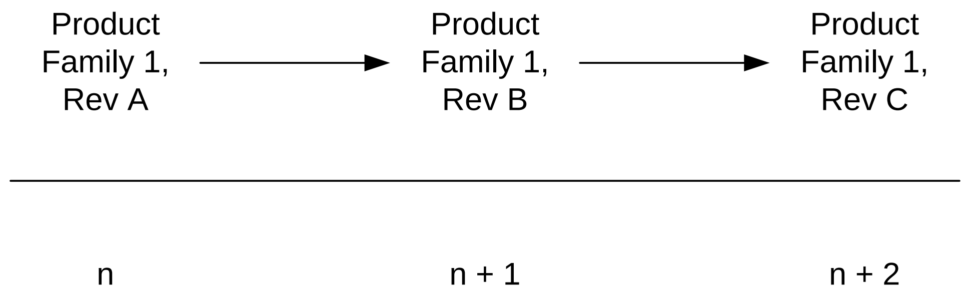

Figure 18. Representation of Product Families and Iteration Numbers

In an engineered-to-order environment, the platform concept is not one that is typically adopted. Instead, these types of companies tend to think in terms of contracts. In order to determine which contracts to collect data from, it is useful to think of these ETO products in a platform context.

By doing so, the fact that products that share components are expected to share similar failure modes, can be leveraged. Data for each contract may be limited, thus the aggregated data from several contracts would allow us to treat them as a platform. Over time, this would allow us to gather reliability data over revisions to the platform (platform iterations) helping to show reliability growth.

Working with an industrial partner, products were grouped into "platforms". These products were grouped by size, time period it was sold in, how it was used, and by the application of the product. In a low volume, high mix product environment, it was a challenge to group products into "platforms" which can then be used during the development of warranty scenarios. It was vital that this phase was completed with the help of the industrial partner as they were the experts in determining what attributes qualify each product to be part of a platform. The benefits go both ways as we were able to bring a fresh perspective to their unique set of warranty problems.

4.2 Warranty Scenarios

The warranty scenario extended the idea of a failure scenario used in an Advanced FMEA (Kmenta & Ishii, 2004). In an Advanced FMEA, the focus is on developing failure scenarios, in contrast to a traditional FMEA which describes the local effect of a component failure. For our purposes, a failure scenario was defined as "an undesired cause-and-effect chain of events", a class of warranty scenarios was defined as a group of similar warranty scenarios, a warranty event was defined as an occurrence of the identified warranty scenario while the product was in operation, and each warranty scenario was defined as a failure scenario that included both diagnosis and repair events (Figure 19).

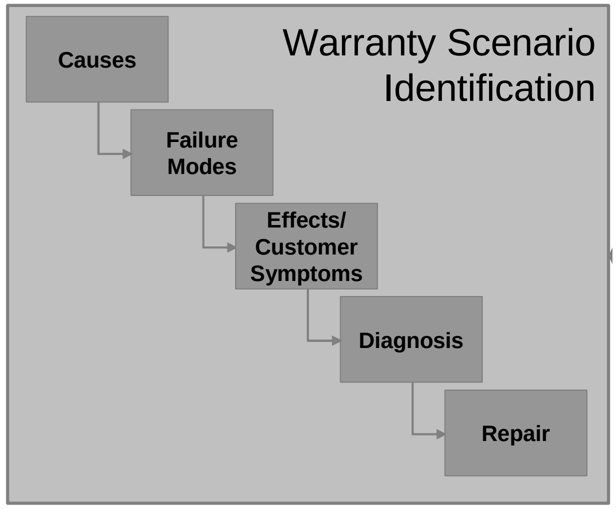

Figure 19. Warranty Scenario Identification (Esterman, et al., 2005)

The first step of this reliability improvement activity was to identify the critical failure modes. When dealing with the development of complex systems, particularly those that incorporate new technologies, a class of failures that caused greatest concern is “unknown-unknown”. These were unanticipated failures resulting from physical mechanisms that were not understood very well. A second class of failure that generated concern were wear-out modes due to the length of time required to uncover them. Both of these could be incorporated in the warranty scenario from the design FMEA. An overview of the steps to developing a warranty scenario is shown in Figure 20.

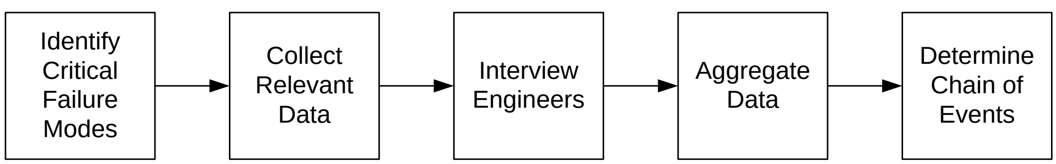

Figure 20. Warranty Scenario Development

Once the critical failure modes were identified, the next step was to determine the causes, effects/customer symptoms, diagnosis, and repair for each failure mode. Traditionally, for a customer facing event, this type of information was stored in the service record or if applicable a failure investigation. It is at this stage that service records were combined with engineering judgment to provide a complete outline of the warranty scenario. This involved interviewing engineers and identifying candidate warranty events that have occurred historically. It should be noted that the warranty scenario includes causes that may be unlikely but can realistically happen in the field.

The development of the warranty scenario was important for two main reasons. It set the groundwork for providing the initial probability of a specific warranty scenario and depending on the

data, its associated cost. Both of these helped to facilitate decision making and provide management an accurate projection of warranty costs so that the team can either have confidence that their actions are leading to improved performance or plan accordingly to the financial impact. In the next section, we will show the data exploration phase of the research. In the end, both the warranty scenario and its related data sets were taken to create an initial probability model.

4.3 Data Exploration Approach

A major piece of this research focused on the identification of data sources that could be used for input into the initial probability model as shown in Figure 21. Using this model, the probability of a warranty event for the current platform iteration could be developed. Therefore, it was critical that an industry partner was identified in order to allow us to uncover additional issues that needed to be solved and therefore help lead to a more robust event generation solution. This was expected to be a challenge as data sources may not be integrated and would mostly likely reside in multiple locations.

Ideally, although the original intent of this research was to be able to integrate data for each platform iteration and compare the probability of each warranty event, this part of the research was never fulfilled due to the complexities introduced by the ETO environment. In the research, only data from one platform iteration was integrated. Therefore, the following steps were taken during data exploration: review of the different types of data available, identifying patterns in data using data mining algorithms, and the use of traditional statistical tools.

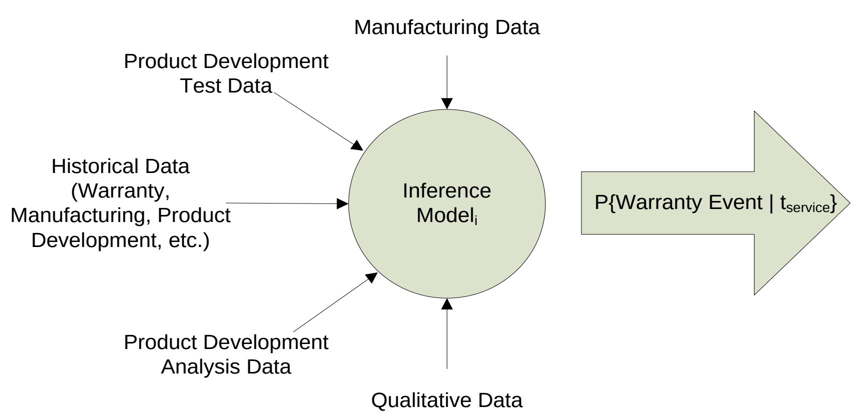

Figure 21. Input-Output Inference Model

A review of available data streams was done with the help of an industry partner. It was determined that engineering judgment could be provided from a variety of tools such as FMEA, fault tree analysis, physics of failure models, and reliability block diagrams. Although most of these tools provided

only qualitative data, this would still be valuable in determining the relative likelihood of particular failures that may be missed when utilizing only quantitative data. The importance of this type of data has increased as more and more products are a combination of mechanical, electromechanical and software components. In the end, we were unable to find any engineering judgment data that could be used as inputs for the inference model.

Prior distribution/historical data were provided from field data records. Although this type of data did not include product revisions already in process, it was important to establish probabilities for previous iterations of the current product platform. It was expected that previous iterations would be helpful in predicting a warranty event for the current platform.

Data mining was used to discover patterns in datasets and thereby gain inferences from multiple sources of data. By identifying underlying patterns, it may be possible to draw unsuspecting relationships and insights of value. This may prove useful when developing platforms in an engineering to order environment.

Specifically, patterns in data (association learning, clustering, etc) were explored using "Weka Explorer". Weka provided an easy to use interface for data mining tasks with its collection of machine learning algorithms. These algorithms could be applied directly to the dataset or used via the user's own custom program. For our research purposes, Weka was used out of the box for its tools on data pre-processing, classification, regression, clustering, association rules, and visualization. It should be noted that the Weka software is also useful for new machine learning schemes which can be added as necessary (http://www.cs.waikato.ac.nz/ml/weka/).

The following data algorithms were applied on the data sources provided by the industrial partner: association learning, clustering and decision trees. Association learning is a way to quantify the relationships between words for qualitative data. Association learning takes each row of data as a string of words and then tries to find a pattern by seeing what words are repeated in each row. Clustering is a way to find subsets of observations that are similar to each other. It is a method of unsupervised learning that allows to user to identify patterns in the dataset. Clustering uses algorithms such as the k-means algorithm which is a simple iterative method to partition a given dataset into a user-specified number of clusters, k (Wu, et al., 2008). The algorithm starts by picking a number of points as the initial k cluster representatives otherwise known as "centroids". This can be done randomly from the dataset or from the global mean of the data. At this point, each data point is then assigned to a cluster by the closest centroid, resulting in a partitioning of the dataset into clusters. Decision trees are effective at predicting the outcome of an event depending on certain pre-conditions. They are set up as a system of rules, similar to

an if-then statement but are usually presented in a tree format. The number of levels as well as the size of the dataset can drive the size of the decision tree. Large datasets can produce decision trees that are quite complicated, creating decision trees that are difficult to visualize as shown in Figure A-6. These data mining algorithms allowed us to better analyze warranty data given the complexity of the interactions between the user, environment, and the product. Allowing automated algorithms and processes aided in the discovery of relevant patterns that allowed us to see connections between causes and undesired states more visibly.

In addition to data mining, linear regression analysis and general linear model was used to determine if warranty events could be predicted effectively using these tools. These are traditional statistical tools that can be used to correlate and model data using linear functions. For example, in order to create a linear regression model to predict warranty costs, the independent variable was set to internal costs and the depend variable was set to field costs. It was expected that these tools would not be successful dealing with the type of data we were expecting but we needed to verify this hypothesis.

Therefore, the data exploration phase of this research started with several traditional sources of data during Product Development: Historical Field Data, Product Development Testing Data, Failure Assessment Tools Data, and Engineering Judgment. Data sources will then be cataloged using a data dictionary which would list the following information for each dataset: attribute, example, description/notes, qualitative/quantitative, type, and length. Field data can then be sorted by cost or frequency in order to prioritize the failure modes. This would help reduce the scope of the problem and give us a feel of where the "big" warranty issues are. It should be noted that relationships between data streams can also be explored with more traditional statistical methods such as linear regression analysis and general linear models. Data mining will then be used to aid in the analysis to model the underlying structures which give rise to consistent and replicable patterns. After analysis, this data can be conditioned and pre-processed so that it is ready to be accepted as inputs into the Bayesian process which will be our initial probability model. This is because a Bayesian framework allows for a systematic process that can accommodate noise, variability, and a lack of data. In the end, this research sets the groundwork for a successful warranty event generation engine that would allow effective warranty prediction early on in the product development process.

The objectives of the data exploration phase was to: identify possible data sources; discover patterns within the data streams; discover relationships between the data streams; link the data sources to the warranty scenarios; develop data conditioning requirements.

4.4 Initial Probability Model

With warranty scenarios identified and data sources linked, an initial probability model was developed. For our purposes, the initial probability model used a Bayesian model based on discrete values. The duration of the warranty period was discovered to vary between one to two years. After reviewing the distribution of mean time to failure data, it was determined that setting the warranty period to one year would provide us the most amount of prediction data.

Initially, the concept of using a Bayesian model based on continuous values was explored. While it has its advantages such as being able to give the probability of a warranty event based on a length of time, it also posed problems given disparate data sets. Thus, a discrete solution was pursued. The inclusion of time was considered but converted to a discrete format. This was done so it would make the model easier to develop in order to demonstrate feasibility. Discrete values would also make data conditioning simpler as use cases could be used to convert number of cycles from a dataset (ex. life test) to an approximate length of time that would integrate well with actual field data. Using continuous values from disparate data sources would add a level of complexity that is not the focus of this particular research phase. It would be difficult to correlate time values from one dataset with another. It is expected that a Bayesian model based on continuous values could be developed in the future.

Therefore, the model developed in this work sought to answer the following question: What is the probability of a warranty scenario occurring in the future within the warranty period? In order to answer this question, the initial probability model relied on using the Bayes equation (Equation 2) as a discrete function.

Probability of event H given evidence E:

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]} \qquad (2)$$

Where
Pr[H|E] = Probability of a warranty event (given a specific warranty scenario) occurring given the warranty period
Pr[E|H] = Probability of the warranty period given the warranty event (given a specific warranty scenario)
Pr[E] = Probability of being within the warranty period (over all events, regardless of warranty scenario)
Pr[H] = Probability of a warranty event (over all time periods, given a specific warranty scenario)

The specific warranty event is identified as H, and E is identified as a time period within the warranty period (<= 365 days). The term Pr [H|E] indicates the probability of a warranty event occurring given the warranty period. This is known as the posterior probability. The term Pr [E|H] indicates the probability of the warranty period given the warranty event. The term Pr [H] indicates the probability of a particular

warranty event over all time periods, regardless of whether it was in or out of the warranty period. The term Pr [E] indicates the probability of being within the warranty period over all events. See Figure 46 for an example of how the warranty events are related to each other in relation to the warranty period and a specific warranty scenario. The term Pr [H|E] that is calculated is the posterior probability, which becomes the prior probability on the next iteration of the Bayesian model as the data is updated with new information.

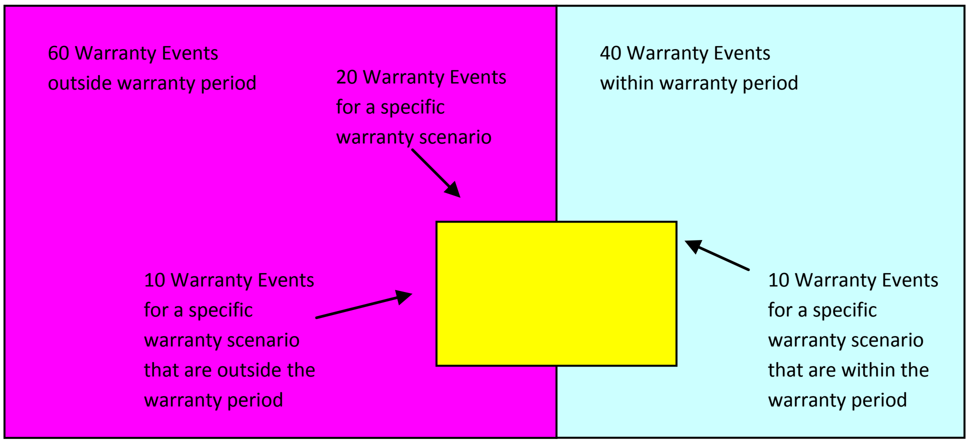

Figure 46. Example of Warranty Events in relation to Warranty Period

From the Bayes equation, we developed the following models for a specific warranty scenario on a given platform. Details on why this approach was taken are explained below:

1. Bayes Approach – Developed a model using one dataset
2. Bayes Approach – Developed a model using one dataset split into two (Training and Test Dataset)
3. Integrated Bayes and Unified Bayes Approach – Developed a model using two datasets. One would be provided by our industrial partner and the other one would be fabricated.

In the first model, we started with a single dataset to show that creating a Bayes model was feasible using product data. For this model, a single dataset, a specific warranty scenario, and a platform were selected. Although we could have used engineering judgment or manufacturing data for our single dataset, we chose historical data because it was easy to quantify.

In the second model, we used a single dataset and split it into two to verify how we could use one part of the dataset to develop a Bayes model and update it using the other part of the dataset (Figure 22). In data mining, data can be split into two groups to verify the accuracy of the data mining algorithm used.

These two sets of data are called the “training set” and the “test set”. We used this data splitting approach to verify how well the Bayesian model reacted to new information. It should be noted that this is not how it is normally used in data mining. In data mining, this type of data splitting is used to check the patterns found from the training set on the test set.

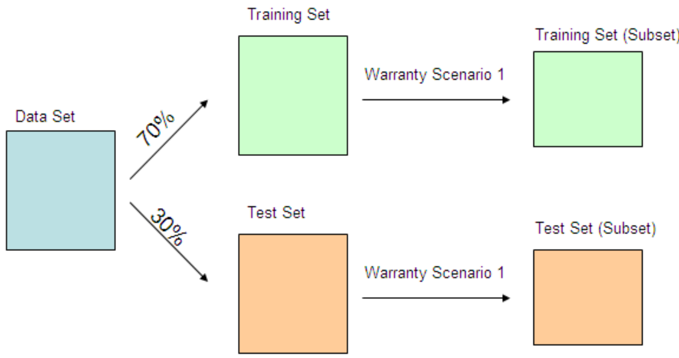

Figure 22. Bayes – Single Dataset (Training and Test Dataset)

In the third model, using the warranty scenarios previously developed, previous historical data (representing platform iteration *n*) and a fabricated life test dataset (representing platform iteration *n+1*), two approaches were used. One approach was the unified Bayes approach (Figure 23) which combined disparate data sets together before input into the Bayes model and the other was the integrated Bayes approach (Figure 24) which created an initial probability model from one dataset and updated the posterior probability with the data from another dataset. The integrated Bayes approach took into account how soon each dataset would be available to the development team through the product life cycle. This usually meant that field data from the previous platform iteration (ex. *n*) would be available immediately for input into the Bayes model whereas the life test data of the current platform iteration (ex. *n+1*)would be available at a later point in time.

The integrated Bayes approach therefore created a new Bayes model for each incoming dataset but updated the posterior probability with the data from the previous dataset. This verified that disparate data sources could be conditioned for input into a Bayesian model. Issues that occurred when qualitative data was integrated with quantitative data as well as different methods to condition available data streams in a Bayesian framework was documented.

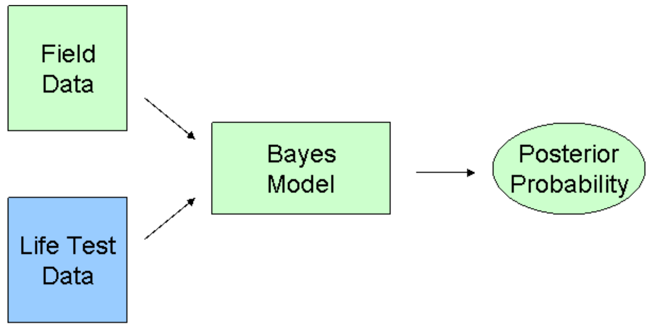

Figure 23. Bayes – Unified Approach

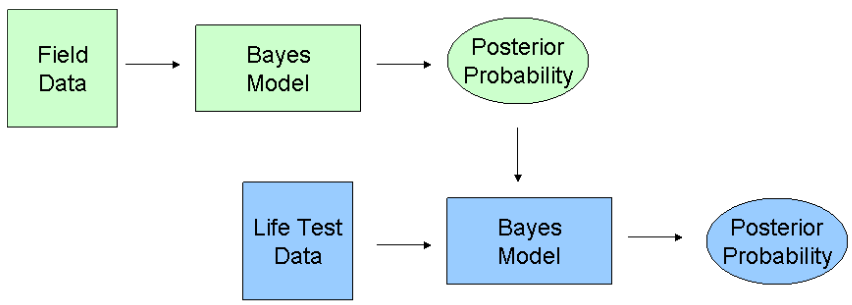

Figure 24. Bayes – Integrated Approach

4.5 Assess Results

Results from all three approaches will be compared with each other to see how much they differ. It will also show how well each approach responds to new data.

4.6 Methodology Summary

As explained in the literature review, Bayesian statistics is a statistical theory and approach to data analysis that provides a coherent method for learning from evidence as it accumulates. Bayesian networks use this theory to update posterior probabilities from prior probabilities. The purpose of this research was one part of a broad goal to successfully utilize a Bayesian approach that would ultimately reduce or eliminate the length of time needed for feedback to product development teams from the field by predicting warranty events. This was important as by the time warranty performance has come back from the field, product development for the next revision may be well under way. Given historical data, it may be possible to utilize Bayesian statistics to predict the likelihood of future warranty events given the likelihood of past warranty events.

5. RESULTS

This research relied on the use of actual data generated by a major large industrial equipment manufacturer. This data allowed us to uncover additional issues that needed to be resolved. With the help of our industrial partner, our initial approach was to group products into "platforms". We then developed warranty scenarios that centered on each platform by interviewing company personnel. We then proceeded to identify data sources that could be used as inputs to the model and created a data dictionary of all data sources used. From that, we did some data exploration based on a cost and frequency approach using field data provided by service records. This was used to prioritize the failure modes we were going to research. Patterns in the data were then explored using Weka. Relationships between data streams were also explored using linear regression analysis and general linear models. Finally, an initial probability model was developed using the warranty scenarios created and the data conditioned for input.

5.1 Platform Concept in an Engineered to Order (ETO) environment

The major industrial equipment manufacturer we worked with focused on providing custom solutions to each customer. Although there were some similarities between products, customers were allowed to configure each individual order to how they wanted. Due to the size of the order, customers could even specify what testing needed to be performed. This presented a unique problem. How to group products in an engineered to order (ETO) environment? This was an important part of the research as products needed to be groups so that data from each product could be aggregated.

In an industry where products were determined by contracts and manufacturing requirements were highly configurable, it was be difficult to identify product families and design platforms. It resulted in a very low volume, high mix product environment. In our research, we found that due to the way products were managed as individual contracts, we needed to find a way to group contracts into "platforms" for this approach to work. Ideally, products could be group into families or platforms and each revision would be an iteration. The challenge of this approach was that it involved multiple interviews with various cross-functional engineering teams to understand how each product was designed so as to provide the basis for each product family.

We developed some ideas on how to group these product lines, shown in Table 26 of how products were similar to each other - whether it was equipment size, technology, application, or the industry of the customer.

Size	Processing	Application	By Industry
A	Linear	Type 1	Industry R
B	Non-Linear	Type 2	Industry W
C			Industry L
			Industry T

Table 26. Table of Product Size, Processing, Application, Industry

Ideally, this may not have been the best approach to identifying platforms for this industry. Martin and Ishii (2002a, 2002b) proposed a two-phase-based QFD method in which the generation variety index (GVI) and coupling index (CI) are used as a measure of the amount of redesign effort required for future designs of the product and the coupling among the product components, respectively. This method can aid companies in developing standardized and modularized product platform architectures.

For our research, due to the uniqueness of the industry, engineering judgment was heavily used to identify and group similar contracts into platforms. During our research we found that although each product was unique, rarely were products designed completely from scratch. Understanding the process of each new contract helped us identify platforms better. After exploring various options to group products, we ended up grouping products by product size.

5.2 Warranty Scenarios

In order to identify events that are similar to each other, we started with a single warranty event and interviewed engineers to understand the full chain of events. This allowed us to explore different paths leading to a warranty event. An example of a class of warranty scenarios that was developed is shown in Figure 27. Each path from left to right is a unique warranty scenario. This was a time consuming process as this type of information was rarely documented and we had to rely on the memories of various service engineers. The time factor limited the number of warranty scenarios that could be developed. Ideally, the design FMEA could be used as a basis for the identification of the warranty scenario and the diagnosis and repair sections can be filled in with input from service engineering.

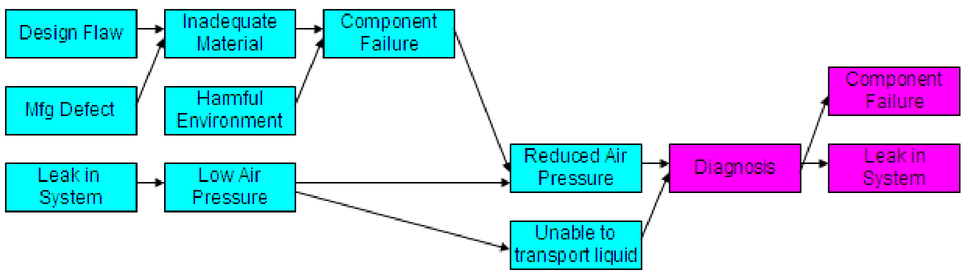

Figure 27. Warranty Scenario Example of Compressor Issue

The development of the warranty scenario was important for two main reasons. It set the groundwork for providing the initial probability of a specific warranty scenario and depending on the data, its associated cost. In the next section, we will show the data exploration phase of the research. In the end, both the warranty scenario and its related data sets can be taken to create an initial probability model.

5.3 Data Exploration

The objectives of the data exploration phase were to: identify possible data sources; discover patterns within the data streams; discover relationships between the data streams; link the data sources to the warranty scenarios; develop data conditioning requirements. In order to accomplish these objectives, a data dictionary was created, warranty events were prioritized, data was explored using Weka, data mining algorithms were applied to the datasets and traditional statistical tools were used to correlate relationships between data streams.

Significant time was spent in the beginning of the research identifying data sources that could be used as input to the model. This was not an easy task as the data was not integrated and resided in various locations. However, by asking engineers to walk through the process of a warranty event, we ended up with a list of data sources. Ultimately, this research settled on three databases: Warranty from the Field Database, Internal Repairs Database, Contract Details Database. Most of the time, data needed to be extracted into a usable format.

A data dictionary (Figure 28) was created to capture the following: attribute, example, description/notes, qualitative/quantitative, type, and length. This helped us to catalog each data source and the format of each field. This information was then used to determine what data mining algorithm could be used on each data source.

Attribute	Example	Qual/Quant?	Type	Length
FIELD NBR	5	Quantitative	Integer	1 to 4
REFERENCE	Contract A	Quantitative	Alphanumeric	5 to 16
ROW ADDED DATE	27-Jul-2001	Quantitative	Alphanumeric	11
CLIENT NAME	Company X	Qualitative	Alphanumeric	6 to 31

Figure 28. Example of Data Dictionary

5.3.1 Warranty Event Prioritization

To reduce the scope of the problem and give us a feel of where the "big" warranty issues were, we used Excel to look at problem code by warranty cost. Problem codes focused mainly on major component groups that failed in the field. In total, there were 95 problem codes with a total cost than ran in the millions (Figure 29). A group of 5 problem codes were selected and then broken down into contributing contracts (See Appendix A-1, A-2, A-3, A-4, A-5). This provided the subset of data that would be used for data mining purposes.

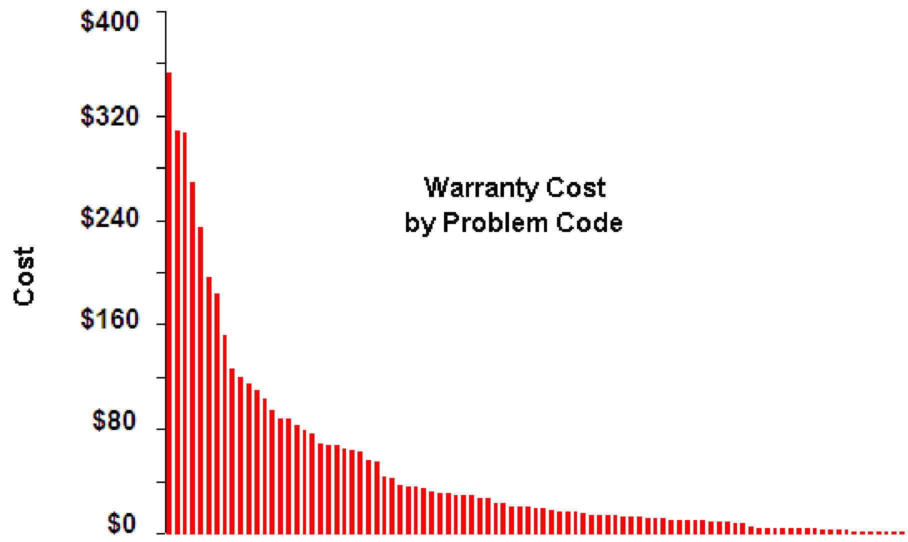

Figure 29. Graph of Warranty Cost by Problem Code

For example, bearings were one of the top contributors to warranty cost. Looking at this data further (Figure 30), we could see that it was driven by a number of contracts; in this case, one of the contracts was a top driving factor. This highlighted the need to look into this contract further.

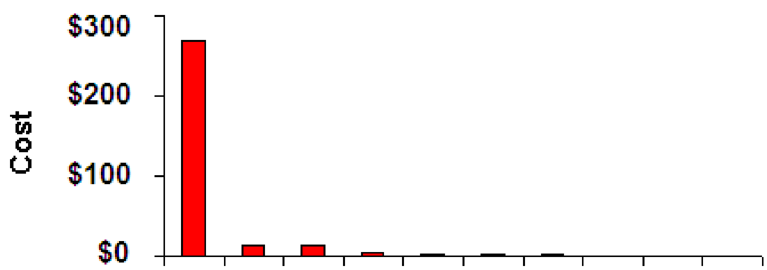

Figure 30. Graph of Bearing Warranty Event Costs by Contract

5.3.2 Time to Failure Histogram

We also looked at the overall time to failure for all products to see where the majority of the warranty events fell time-wise. As expected, most of the warranty events occurred under two years. This coincided with our knowledge that the standard warranty offered by our industrial partner was one or two years. An observation we made looking at the failure data was that the actual field data does not represent a true bathtub failure curve (Figure 31). In a theoretical bathtub curve, there would be a period of decreasing failure rate soon after a product is launched, followed by a constant failure rate before increasing again as the product ages. Long lead times for parts and pressure to meet delivery dates

(penalties included in contract) may be one of the many causes for a high failure rate at the beginning of the product life.

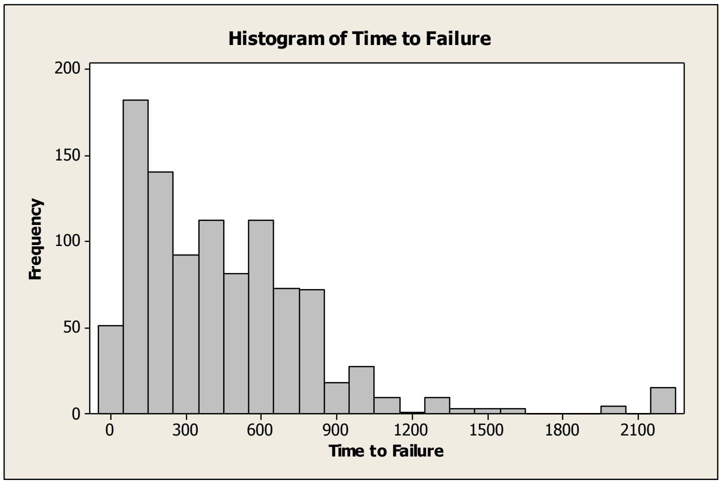

Figure 31. Histogram of Time to Failure for all Products

5.3.3 Data Mining using Weka Explorer

Patterns in data (association learning, clustering, etc) from the data sources provided by our industrial partner were explored using "Weka Explorer" (Figure 32). Weka Explorer was used to analyze the data and it provided information such as number of attributes, number of instances, histogram of selected attribute, number of instances per class for each attribute, and percentage of unique values per attribute. We hoped to find similar warranty events that could be grouped into warranty scenarios as well as explore relationships between the data sources.

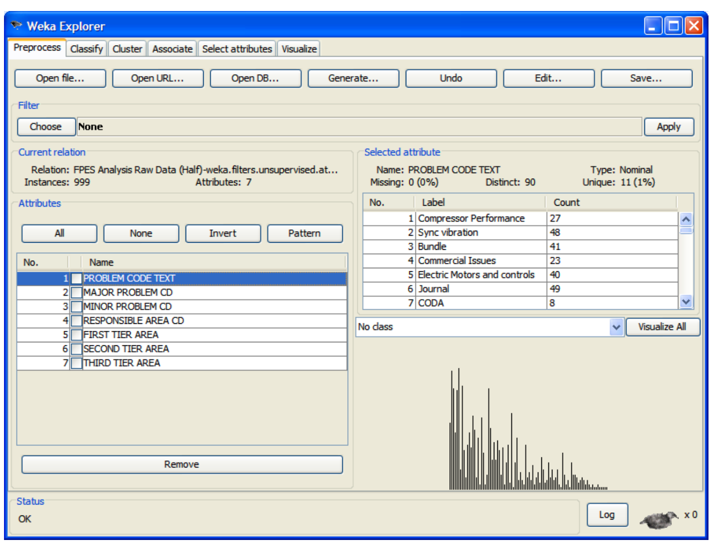

Figure 32. Weka Explorer Graphical User Interface

In order to use Weka, the data (originally in an Excel file format) was converted to a CSV (comma-separated values) file format. The CSV file format is a simple text format for a table. In it, each row in the dataset is converted to one line in the text file where each parameter is separated by commas or semi-colons.

We used Weka to analyze this data, providing information such as number of attributes, number of instances, histogram of selected attribute, number of instances per class for each attribute, and percentage of unique values per attribute.

Although it was easy once we got going, it was not without its issues. The file conversion took longer than expected, and certain fields had to be removed in order for the file to be converted to a CSV file format such as: problem cause, correction, and description (long text). In addition, due to the large dataset used, we ran into memory allocation errors forcing us to select a subset of our data. Large datasets also made visualization of decision trees and clustering difficult due to the sheer number of levels in each attribute.

5.3.4 Association Learning using Weka

We used association learning as a way to quantify the relationship between words for qualitative data sources. This was done by determining how often two words appear together. Based on our data dictionary, we picked a qualitative data source and the field "cause of warranty event" (Table 33) as the scope of the association learning algorithm. Although this type of data was inherently noisy due to spelling mistakes and consisted of numerous inconsistencies, it presented the opportunity of interesting results. We tried to find relationships between words in the "cause of warranty event" field in order to develop a method of grouping similar warranty events together.

Cause of Warranty Event
When the component was being removed, the seals were not properly tagged and stored for reassembly.
Carbon steel housing was selected for pre-specified application.
Most likely we have damage to the primary seal. Root cause is unknown at this time.
Several of the components on the Contract A assembly were significantly out of tolerance from their axial locations. This problem was not discovered until the unit was being assembled at the customer's site during the turnaround (Ref. Service Record 1). In order to ensure that this problem does not occur during the upcoming turnaround for contract A, the customer wants manufacturer to comfirm the axial dimensions on the assembly.
Internal recycle due to excessive internal clearances. Clearances on component build drawing are in error.
Issues with controls and motor support structure [assumption as investigation is underway].

Table 33. Example of Qualitative Data used in Association Learning

This was accomplished by inputting each row of data as a string of words into the data mining algorithm. The association learning algorithm then tried to find a pattern by seeing what words was repeated together in each row. An example of the user display for the association learning interface for Weka is shown below in Figure 34 and the results are shown in Table 35.

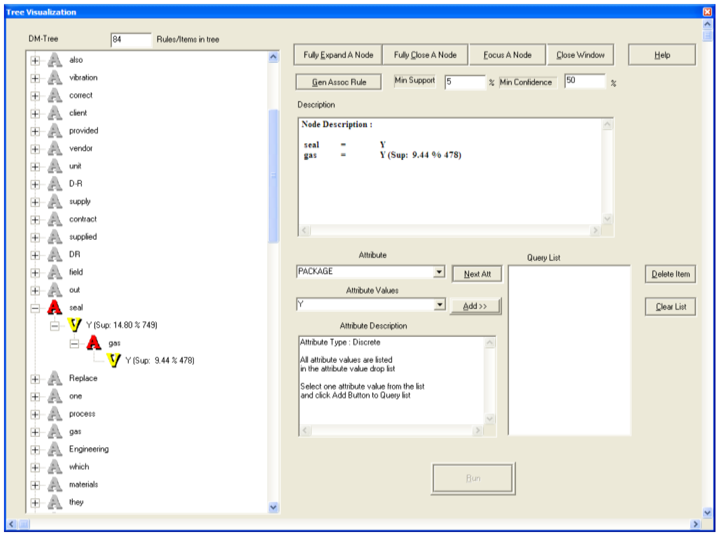

Figure 34. Example of User Interface for Association Learning

Results	
Item 84:	Item 83:
gas = Y	Company = Y
seal = Y	site = Y
(9.445% 478)	(6.283% 318)

Table 35. Association Learning Results

From the results given above (Table 35), the two highest level of association was Item 84 and Item 83. In item 84, the 9.445% indicates the support and the 478 indicate the number of occurrences. Our use of association learning did not prove successful as it did not reveal any new information that we did not previously know.

5.3.5 Clustering using Weka

We used clustering in Weka to find observations that may be similar to each other. In Figure 36, these observations are shown in color to indicate subsets of observations. We used clustering on the field dataset provided by our industry partner and were able to partition the dataset into clusters. This was done to find clusters of common events. We did not find any of these clusters to be meaningful.

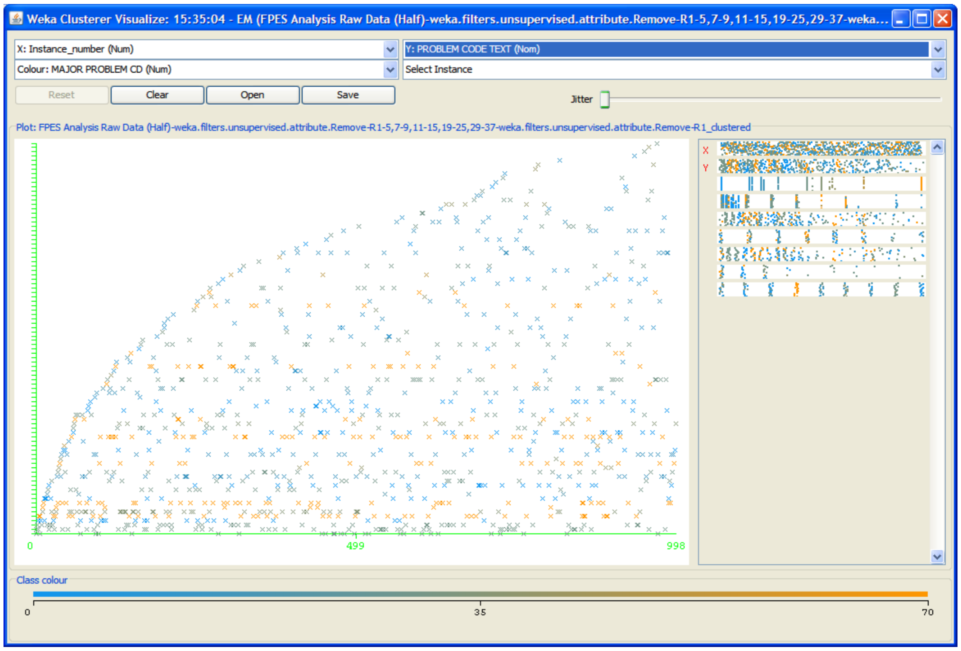

Figure 36. Weka's User Interface for Clustering

5.3.6 Decision Trees using Weka

We used decision trees to try to predict the outcome of an event depending on certain pre-conditions. These rules were setup in a tree format that could be displayed. In our research, we first used an attribute selection tool (Figure 37) to select the "best" attributes and then a classifier tool to set up our decision tree. The first tool we used analyzes the attributes depending on the class chosen to indicate which attributes will give the "best" results. In our research we found out that the Attribute Evaluator and Search Method chosen thinks that the best attributes to go with "Problem Code Text" is "Row Added Date", "Major Problem Code", "Minor Problem Code". Selecting a different class "Total Actual" results in a whole other set of best attributes: Field Service #, Field Service Amt, Material Amt Sum, No Charge Matl Est Amt Sum. Weka also includes a variety of search methods that can be used to build the decision tree. For our purposes, the BestFirst algorithm was used. BestFirst algorithm searches a graph by choosing the most promising or "best" node according to a specified rule. (Research BestFirst and other algorithms available on Weka). The classifier we used was J48, which is WEKA's C4.5 implementation.

In general, decision trees are helpful in conditioning future data once the rules are set from the beginning. Our data does not seem to have enough of this type of data for this to be effective for us.

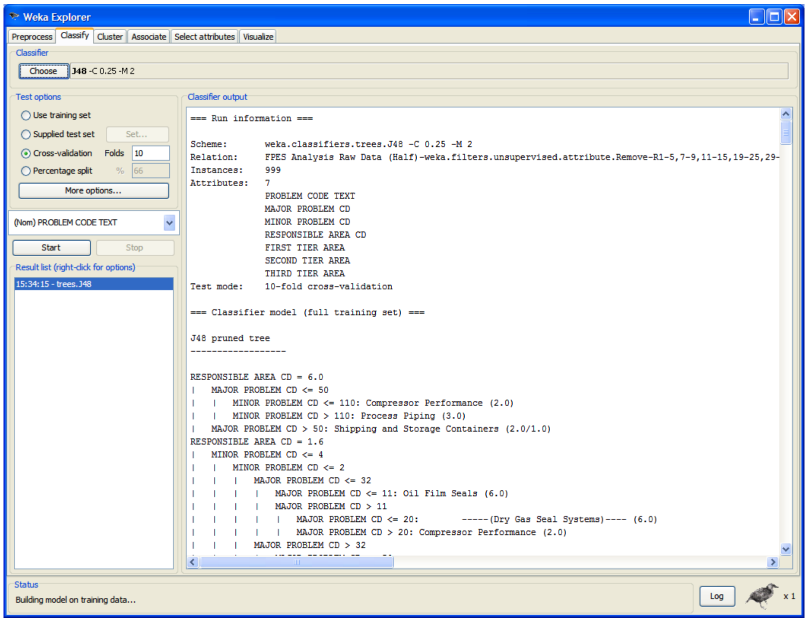

Figure 37. Weka's User Interface for Decision Trees

5.3.7 Discovering Relationships between Data Streams using Linear Regression Analysis

We used linear regression analysis and general linear models to see if these tools could be used to predict warranty events. In order to do this, we analyzed how the various databases related to each other using linear regression analysis. We wanted to understand how costs incurred during the manufacturing of a contract had an effect on the cost of repairs done in the field (post production).

We set the dependent variable Y as "Field Cost" (Cost incurred post production provided by field service data) and the independent variable X as "Internal Cost" (Cost incurred during manufacturing provided by internal repair data). Our null hypothesis was that the slope is equal to zero (no significant linear relationship between independent variable X and dependent variable Y). We ended up with the following equation and output from Minitab (Table 38a and Figure 38b):

Regression Equation					
Field Cost = 27343 + 0.308 Internal Cost					
Predictor	**Coef**	**SE Coef**	**T**	**P**	
Constant	27343	12968	2.11	0.037	
Internal Cost	0.3075	0.2548	1.21	0.229	
S=140884	R-Sq=0.9%	R-Sq(adj)=0.3%			
Analysis of Variance					
Source	DF	SS	MS	F	P
Regression	1	28920097365	28920097365	1.46	0.229
Residual Error	159	3.15586E+12	19848203424		
Total	160	3.18478E+12			
Unusual Observations					
17, 101, 103, 108, 110, 125					

Table 38a. Minitab Output of Regression Equation for Field Cost and Internal Cost

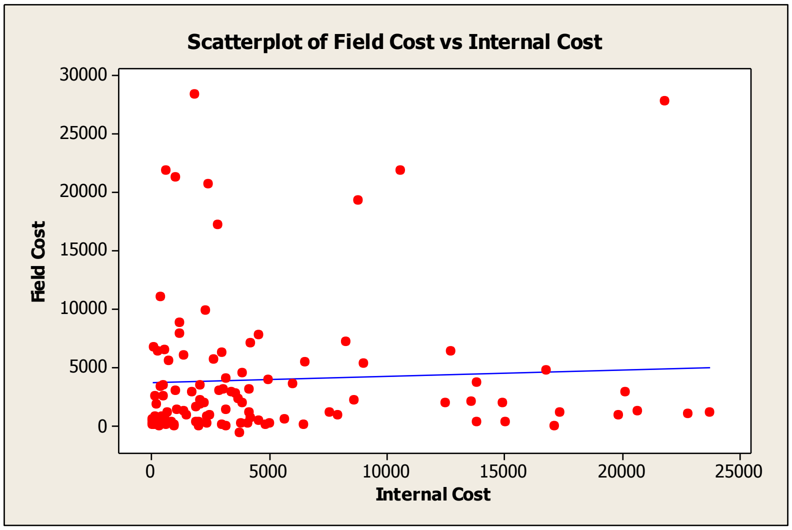

Figure 38b. Scatterplot (Minitab Output) of Field Cost (Y-Axis) vs. Internal Costs (X-Axis)

From the output (Table 38a), the regression equation explained 0.9% of the variation in Y that can be attributed to X. This was a low value that suggests the model was a poor one for the data set. A higher percentage would indicate that the model was a good one for the data set. The p-value of Internal Cost was 0.229. Since this was much greater than 0.05, we could not reject the null hypothesis. This indicated that there was no relationship between Internal Cost and Field Cost. There were also several large values for standard residuals for a few observations (17, 101, and 103). In the future, it is possible that we may want to throw these values out and refit the model.

As linear regression analysis assumes normality, we tested the normality assumption using Kolmogorov-Smirnov. Looking at the data, it appeared neither Internal Cost (Figure 39) nor Field Cost (Figure 40) resembled a normal distribution. This made sense as it helped to explain possible reasons to why our linear regression equation was not a good model for the dataset and why Internal Cost was not a good predictor of Field Cost.

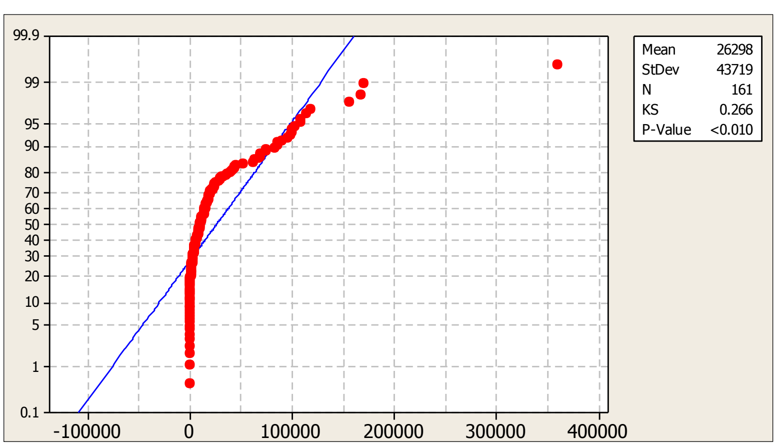

Figure 39. Probability Plot (Minitab Output) of Internal Cost (X-Axis) vs. Percent (Y-Axis)

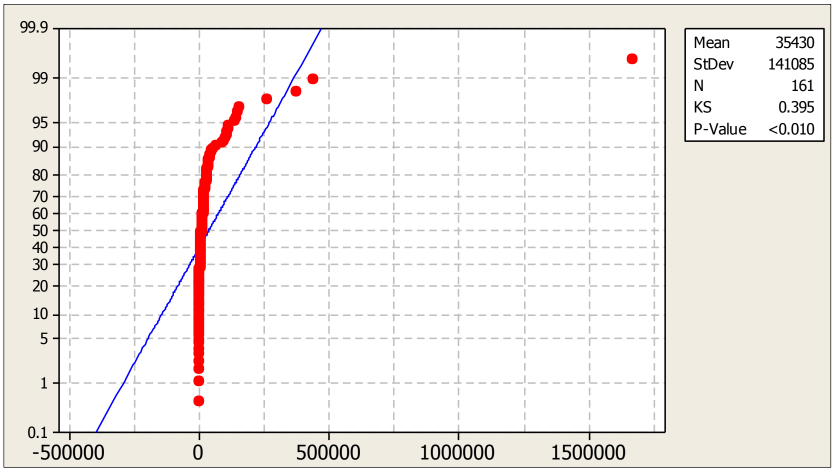

Figure 40. Probability Plot (Minitab Output) of Field Cost (X-Axis) vs. Percent (Y-Axis)

We repeated our analysis using the number of events that occurred during manufacturing and the number of events that occurred in the field (post production).

We set the dependent variable Y as "Field Count" and the independent variable X as "Internal Count", see Figures 41a and 41b. We wanted to find out if the number of repair events that occurred during manufacturing was a good predictor of the number of repair events that occurred in the field (post production). Our null hypothesis was that the slope was equal to zero (no significant linear relationship between independent variable X and dependent variable Y).

Regression Equation					
Field Count = 4.44 + 0.0837 Internal Count					
Predictor	**Coef**	**SE Coef**	**T**	**P**	
Constant	4.439	1.118	3.97	0.000	
Internal Cost	0.08369	0.04613	1.81	0.072	
S=8.20117	R-Sq=2.0%	R-Sq(adj)=1.4%			
Analysis of Variance					
Source	DF	SS	MS	F	P
Regression	1	221.39	221.39	3.29	0.072
Residual Error	159	10694.21	67.26		
Total	160	10915.60			
Unusual Observations					
13, 17, 36, 103, 132, 135, 156					

Table 41a. Minitab Output of Regression Equation for Field Count and Internal Count

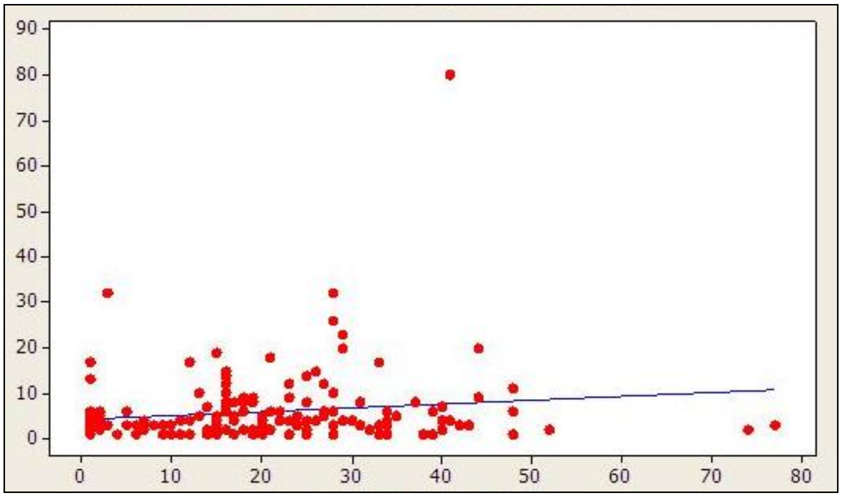

Figure 41b. Scatterplot (Minitab Output) of Field Count (Y-Axis) vs Internal Count (X-Axis)

Looking at the output, the p-value of Internal Cost was 0.072. Since this was greater than 0.05, we could not reject the null hypothesis. This indicated that there was no relationship between Internal Cost and Field Cost at this confidence level. The R-Sq was 2.0% which was very low value. This indicated that the model was a poor one for the data set. Several observations (13, 17, 36, and 103) had large standard residuals, which indicated that we may want to throw these out and refit the model.

We tested the normality assumption using Kolmogorov-Smirnov. Looking at the data, it appeared that the Field count data was not a normal distribution (Figure 42a). Internal count data appeared to better fit a normal distribution (Figures 42b).

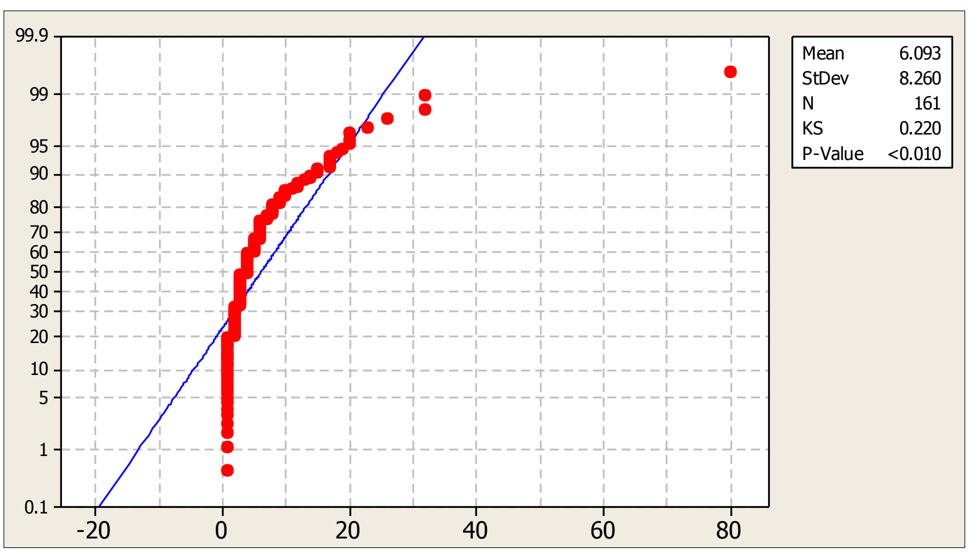

Figure 42a. Minitab Output of Probability Plots of Field Count (X-Axis) vs. Percent (Y-Axis)

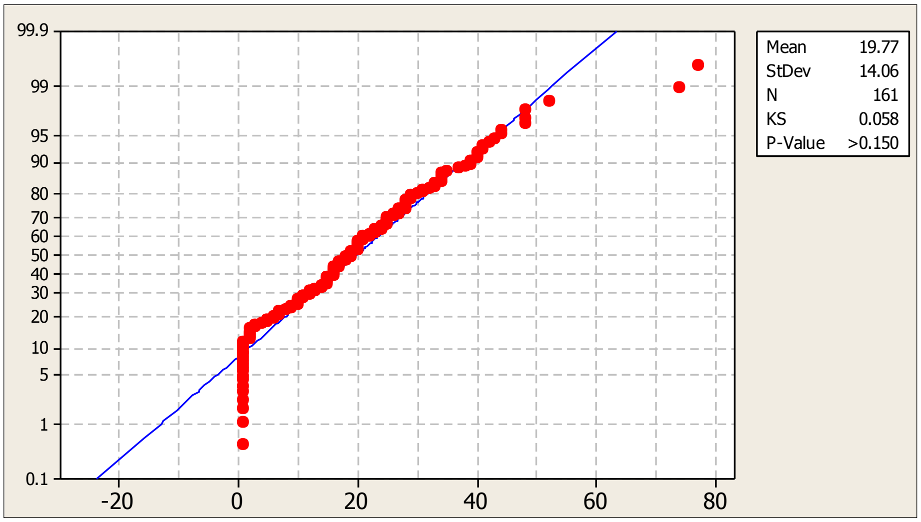

Figure 42b. Minitab Output of Probability Plots of IR Count

5.3.8 Discovering Relationships between Data Streams using General Linear Model

The general linear model could be seen as an extension of linear multiple regression for a single dependent variable. Although similar to a linear regression approach, the general linear model goes a step beyond the multivariate regression model by allowing for linear transformations or linear combinations of multiple dependent variables. This extension gives the general linear model important advantages over the multiple and the so-called multivariate regression models, both of which are inherently univariate (single dependent variable) methods. (http://www.statsoft.com/textbook/general-linear-models/).

For our purposes, our goal was to find relationships between the response and the model using general linear models. In order to show this, we needed a model that had a low p-value but a high R-sq value. Our conclusion was that we were unable to find a GLM model which strongly correlated the response and model. Results are tabulated in Table 43.

Response	Model	Levels	p Value	R-sq
Field Count	Major Problem Code	11	0.413	9.91%
Field Count	Minor Problem Code	33	0.205	35.63%
Field Count	First Tier Area	8	0.577	5.50%
Field Count	Product Code	3	0.498	1.34%
Field Cost	First Tier Area	8	0.000	1.91%
Field Cost	Minor Problem Code	47	0.000	4.89%
Field Cost	Major Problem Code	13	0.000	4.23%
Field Cost	Product Code	3	0.366	0.11%
Time to Failure	Field Count, Field Cost	20	0.006, 0.331	22.42%
Field Cost	Part Noun	63	0.961	46.94%
Field Cost	Frame Size	10	0.814	5.12%
Field Cost	Internal Cost, Internal Count Code	4	0.879, 0.236	4.39%
Field Cost	Time to Failure, Internal Count	18	0.731, 0.717	81.46%
Field Cost	Time to Failure, Field Count	11	0.246, 0.000	96.36%

Table 43. Table of results for GLM

In conclusion, although we were able to successfully apply data mining techniques and traditional statistical methods, we were unable to find any useful underlying patterns or relationships between data streams. Nevertheless, we were able to focus on the major problem codes and identify what data streams could be used as inputs to the model.

5.4 Initial Probability Model

After verifying our hypothesis that traditional statistical methods were insufficient in being used to effectively predict warranty events, an initial probability model was developed that would better handle the available data streams. Developing this initial probability model involved the following three approaches: developing a model using one dataset, developing a model using one dataset split into two, and developing a model using two datasets.

5.4.1 Bayes Approach – One Dataset

Our initial probability model relied on using the Bayes equation (Equation 2) as a discrete function for a single dataset. This approach used only the historical dataset (field data) and six contracts grouped as warranty scenario 1.

Initially, we set Pr [E|H] as the count of events for a particular warranty scenario within the warranty period over the count of all events for a particular warranty scenario. Pr [H] is set to the count of events for a particular warranty scenario over all warranty events. Pr [E] is set to the count of all events within the warranty period over the count of all warranty events. When we put this in the equation, this gives us the probability of a warranty scenario occurring given that it is within the warranty period.

$$\Pr[H] = \frac{81\, \textit{events for a warranty scenario}}{1007\ \textit{warranty events}} = 0.0804$$

$$\Pr[E] = \frac{489\, \textit{events within warranty period}}{1007\ \textit{warranty events}} = 0.4856$$

$$\Pr[E \mid H] = \frac{27\, \textit{events for a warranty scenario within warranty period}}{81\, \textit{events for a warranty scenario}} = 0.3333$$

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]} = \frac{(0.3333)(0.0804)}{(0.4856)} = 0.055$$

Pr [H|E] = Probability of a warranty scenario occurring given that it is within the warranty period

Pr [E|H] = Probability of an event occurring within warranty period given a particular warranty scenario

Pr [H] = Probability of a particular warranty scenario occurring

Pr [E] = Probability that any event is within the warranty period

This initial probability model showed that the probability of a warranty scenario occurring given that it is within the warranty period is 0.055. We would compare this with the results from the other approaches.

5.4.2 Bayes Approach – One Dataset (Training and Test Dataset)

Next, we split the data into two groups. These two sets of data were called the "training set" and the "test set". This type of data splitting was used to verify how well the Bayesian model reacted to new information.

We took our entire dataset of 1,007 entries and split it into the following groups: 30% (302 entries) as Test Set and 70% (705 entries) as Training Set. Next, we took a warranty scenario and created a Bayes model. We started with Sync Vibration as Warranty Scenario 1 and took the 705 entries previously identified as the Training Set and found which entries corresponded to Warranty Scenario 1.

For our initial Bayes model, we had the following

P(E|H): 0.266666667

P(H): 0.063829787

P(E): 0.524822695

This gives us:

P(H|E): 0.032432432

We then took the 302 entries previously identified as the Test Set and found which entries correspond to Warranty Scenario 1. This data helped us to update the initial probability model which resulted in the following:

P(E|H): 0.416666667

P(H) is the P(H|E) from previous model: 0.032432432

P(E): 0.400662252

This gives us:

P(H|E): 0.033727943

It was interesting to note that in the first approach, the probability obtained was 0.055 compared to 0.033727943 obtained in this approach. Since both of these approaches used the same dataset and focused on the same warranty scenario, we analyzed why these numbers were different. We concluded that when the dataset was split, the test set had a much smaller probability than the training set, indicating that the frequency of a particular warranty scenario was going down.

5.4.3 Integrated Bayes and Unified Bayes Approach – Two Datasets

Our initial probability model used a single dataset to develop a posterior probability which proved successful. We wanted to see how a model would deal with two different datasets, and decided on using two different approaches in doing this.

In the first approach, the Integrated Approach, we created an initial Bayes model from one dataset and using the posterior probability of this dataset, combined it with the Bayes model from the other dataset. In the second approach, the Unified Approach, we combined both data sets together and created our Bayes model from ne combined dataset. As expected, in the unified approach, a weighting factor resulting from the size of each dataset had an effect on the probability.

For both approaches, we used two different datasets: life test data and historical data. This was done to show that multiple data streams could be conditioned for input using the Bayes' theorem. In order to condition both data sets for input into the model, we converted the life test data from number of cycles to an approximate length of time. This allowed us to discretize this value into whether it was in or out of the warranty period.

Approach	P(E|H)	P(H)	P(E)	P(H|E)
Unified Bayes (2 Datasets)	0.396	0.106	0.487	0.086
Integrated Bayes (2 Datasets)	0.566	0.032	0.525	0.035

Table 44. Table of Probabilities from Unified and Integrated Bayesian Approaches

Both approaches provided the product developer a method of using historical data and current testing data to provide feedback to product developers whether or not their actions are leading to improved warranty performance in the field.

There seemed to be a significant difference between the posterior probability obtained using the unified Bayes approach (0.086) and the posterior probability obtained using the integrated Bayes approach (0.035), tabulated in Table 44. This is mainly due how each data set was weighted in the model. In the unified Bayes approach, both data sets were combined into one data set before it was inputted into the model. Since the life test dataset was smaller than the field dataset, its weight was correspondingly smaller. However, in the integrated Bayes approach, both data sets were given equal weights. This was because the initial probability model was created using the field dataset and updated using the life test dataset.

5.4.5 Data Conditioning Requirements for Disparate Data Sources

The incorporation of different data into the Bayesian model was not a simple task. In order to integrate disparate data sets into one dataset, the same set of parameters needed to be used. In the initial probability model, life test data was used to demonstrate how this data could be integrated. Due to the lack of disparate data sets available from the industry partner, this life test data was generated.

The generated life test data included 30 samples for each warranty scenario with failures at a random number of cycles, shown below in Table 45. The first two columns on the left are representative of what actual life test data would look like in industry. In order to prepare this data for integration, two columns on the right were added. The first column added, “failure mode” ties each sample to a particular failure mode. The second column added, “> 365 days” converts the number of cycles (based on a use case scenario) to a time period that would represent whether the sample failed in or out of the warranty period. By adding these two columns, this converted the life test data into a usable format that was similar to the parameters already set by the field data.

Sample #	# of Cycles	Failure Mode	> 365 Days
1	809	Warranty Scenario 1	Yes
2	340	Warranty Scenario 1	No
3	58	Warranty Scenario 1	No
4	568	Warranty Scenario 1	Yes
5	463	Warranty Scenario 1	No
6	107	Warranty Scenario 1	No

Table 45. Subset of Life Test Data

A rough process was sketched out to condition each dataset so they could be prepared for input into the initial probability model. It resulted in the following steps:

1. Determine Time to Failure for each contract
    a. Use customer delivery data to find out when product was shipped
    b. Use field data to find when product first had issues (first warranty event)
2. Determine warranty period
3. Link warranty events to platforms previously identified by doing the following
    a. Gather product development data
    b. Interview engineers
4. Input into discrete probability Bayes model

By conditioning each dataset into a single format it was possible to combine two different types of data into one. This was performed for both the unified and integrated Bayesian approaches verifying the feasibility of conditioning available data streams for a Bayesian framework.

6. DISCUSSION

The goal of this thesis was to develop a mathematical framework to integrate Bayesian methods and data mining techniques to develop the event generation engine in an engineer to order environment. The original objectives were: Can multiple data streams be conditioned for input using the Bayes' Theorem? What are some of the issues that can occur when qualitative data is integrated with quantitative data? What are the different methods to condition available data streams in a Bayesian framework? These were questions that had not been answered in current literature with regards to the integration of data in a Bayesian framework for warranty prediction.

The original objectives were partially accomplished by integrating two data streams (field and life test data) and conditioning it for input into a Bayesian model to predict event rates. Three different approaches were taken to create the initial probability model using Bayes' theorem. This resulted in different posterior probabilities which may be attributable to the size of the datasets used. However, by showing that two disparate data sources can be integrated and that this can be used to predict event rates, this research was successfully in showing feasibility of this approach.

Ideally, multiple disparate data sources would have been used to show complete success of the original objectives. This was not possible due to the limited number of datasets provided by the industry partner in this research. Nevertheless, the initial probability model that was obtained in the results section can be used to predict the frequency of a warranty scenario. This means that we know what the probability of that particular warranty scenario in the future.

Knowing the probability of a warranty event is one part of the framework for predicting warranty performance shown in Figure 8. Using this, it is possible to develop another model to generate warranty scenario costs which will ultimately be used in prioritization and risk mitigation during the product development process.

6.1 Challenges

Although this research provided a simple Bayesian model for predicting warranty scenarios, there were numerous challenges encountered which presented some limitations to the research.

Learning Curve: There was a learning curve involved in dealing with data from an engineered-to-order environment. Complaint data was contract based which was in contrast to the author's background in market based environment. This limitation was overcome with the development and use of the platform concept.

Data Gathering: Data gathering proved difficult as there were many data entries that were noisy or incomplete. The industry partner did not use an enterprise resource planning (ERP) system which made finding all the relevant data complicated. For example, risk documents were on multiple locations on the network and were not consolidated in one location. In addition, variations in the naming of the same contract or product made grouping warranty events into warranty scenarios difficult. There were also a large number of null values in the database which required manual data cleanup.

Cost vs. Frequency: One of the challenges involved in developing warranty scenarios was that the high cost warranty items were more often than not "one-off" events. Focusing on the warranty cost as a discriminating factor may not have been the best approach. This is discussed in the conclusion and future work section.

6.2 Conclusions

This research proposed a method to condition available data streams for use in a Bayesian framework. An initial probability model was developed using the Bayes' Theorem with actual data provided by an industry partner in an ETO environment. It was also shown that two different data streams can be integrated to show feasibility of this approach. This type of information can prove useful to product development teams early on in the product development life cycle. Furthermore, by working in an ETO environment, the challenges associated with that environment were also identified.

Platform Identification in an ETO environment was explored in this research. We were able to successfully group contracts into "platforms" for our approach to work. By understanding the product development process, we were able to identify similar contracts into platforms. This rough outline of how platforms are identified can be used in future research with ETO companies.

This research developed warranty scenarios from interviews with engineers and field data. These warranty scenarios were useful in understanding not only the cause-effect-diagnosis-repair of the warranty event but also helped to identify where each data stream was located in the process. This could be particularly helpful with the development of complex systems, especially those that incorporate new technologies. Although useful, they were also time intensive to create as it involved numerous interviews with various engineers. It is possible that this process could be made more efficient or a system could be in place where this type of information is taken from when the first customer call is made.

Data was explored in many different ways for this research. Although some of the approaches did not prove successful, it enabled us to better understand the data. For example, data mining was one approach that held much promise in uncovering patterns of data that were not previously identified. In reality, this type of approach proved to be quite involved as it involved many hours of manual data cleanup and ended up unfruitful in results. It is possible that our approach for data mining could have been better planned out as clustering could have been used to prioritize the subset of complaint data to analyze. Instead we used Excel to prioritize by warranty costs which may not have been the best approach. It would have been interesting to see if clustering would present groups of small warranty events that collectively resulted in a high warranty cost.

The initial probability models developed provides the management team an insight into the probability of a future warranty event that is similar to one that happened in the past. Three models were developed to show feasibility of this approach. The first model used a single dataset to show the initial feasibility of the Bayesian approach. In the second model, we used the same dataset but split it into two

parts so we could see how well the Bayesian model reacted to new information. Both the first and second models were used as building blocks for the third model which used two different datasets to develop a Bayes model. This showed how well the model dealt with two different datasets. In the future, it would be interesting to see how well the Bayes model incorporate data from many different data sources.

Predicting warranty performance during the product development phases involves providing the management team an accurate projection of warranty costs so that the enterprise can plan accordingly. These impacts include, product pricing, extended warranty support pricing, service inventory requirements, warranty accruals, etc. Although an easy way to find out warranty costs is to take the average cost of past warranty events and multiply this by the probability of a warranty event occurring, more granularity is necessary in order to provide details on the type of impact. An alternative approach could be to use each independent warranty scenario and characterize not only its probability of occurrence but also its cost. These two characteristics can be used to calculate an expected cost for each warranty scenario.

This research has the potential to reduce warranty costs for companies. With shrinking margins, warranty costs are one way to increase gross margins and improve customer satisfaction. Understanding the impact of design changes on warranty costs will help product development teams design better quality products. This potential benefit provides the argument for further research into the use of Bayesian methods and data mining techniques for development of the event generation engine.

6.3 Future Work

The work accomplished in this research is only the beginning of the development of a system for warranty prediction during product development. In this research, a simple discrete model using the Bayes' Theorem was used. The device time to failure was converted to "yes" if it failed within the warranty period or "no" if it did not. It is possible to use a continuous model of the Bayes' Theorem. This would represent time in a more accurate fashion as it is inherently a continuous variable. Implementation of this approach would be more difficult however, as it would have to deal with mix models of data. In addition, depending on the industry it may not be easy to calculate traditional reliability characteristics such as mean time to failure. For example, while it is common for retail customers to use a particular product soon after they buy it, companies that operate in remote parts of the world purchase large equipment in pairs and leave one to sit as a spare for years. This is particularly evident in industries where downtime is a primary concern.

Feasibility of this approach was accomplished in this research. The next phase should focus on validation of the model. This can be accomplished via a retrospective case study or a predictive case study (Esterman, et al., 2005). For the retrospective case study, a past development project with a complete set of product development data and stable field data will be examined in order to test the models and develop insights into their strengths and weaknesses. Following this activity, the true test of the methodology will be to apply it to a product under development. The key will be to structure a monitoring process to determine if the methodology aided product developers in the decision making process. If the team feels that they gained insights that they would not have gained otherwise, then that will also be deemed a success for the process.

Warranty scenario development was explored in this research. It was found that this was a time consuming process that presented numerous opportunities for improvement. Although we used warranty scenarios to develop our Bayes model, we did not explore developing Bayes models for a specific chain of events in a given warranty scenario. This may not be possible with limited data, but it is an opportunity for future research.

Additional opportunities for future work include the weighting of multiple data sets has not been fully explored in this research. Although it has been shown how weighting can affect the posterior probability from the two different Bayesian approaches proposed, the size of the dataset may not be the optimal approach to determining the weight of a specific dataset. With the addition of more datasets (e.g. engineering judgment data), this issue will become more prominent. A method for determining the weight

of each dataset will need to be developed in the future. This will help maximize the utility of data streams that exist within the enterprise.

7. APPENDIX

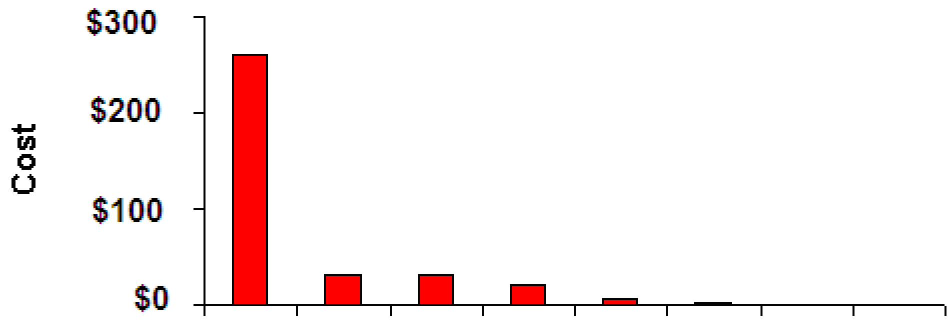

Figure A-1. Cost Breakdown of Mechanical Wear Warranty Events by Contract

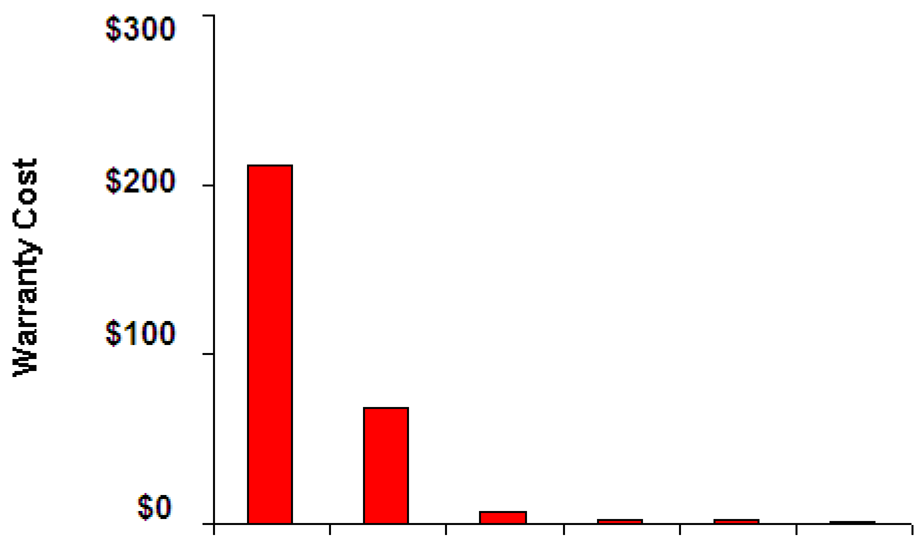

Figure A-2. Cost Breakdown of System Failure Warranty Events by Contract

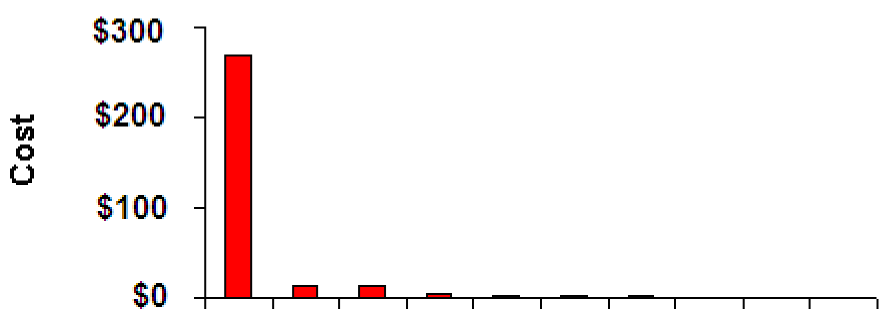

Figure A-3. Cost Breakdown of Mechanical Component Warranty Events by Contract

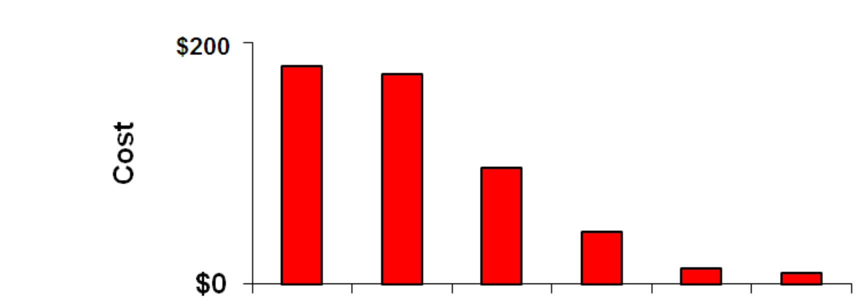

Figure A-4. Cost Breakdown of Electrical Component Warranty Events by Contract

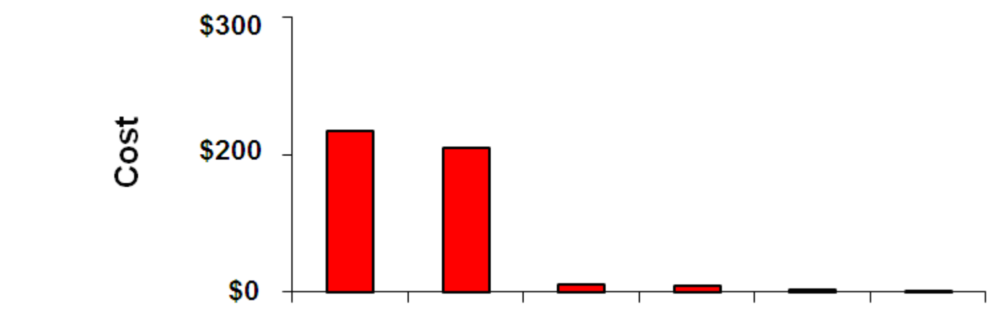

Figure A-5. Cost Breakdown of Miscellaneous Issues Warranty Events by Contract

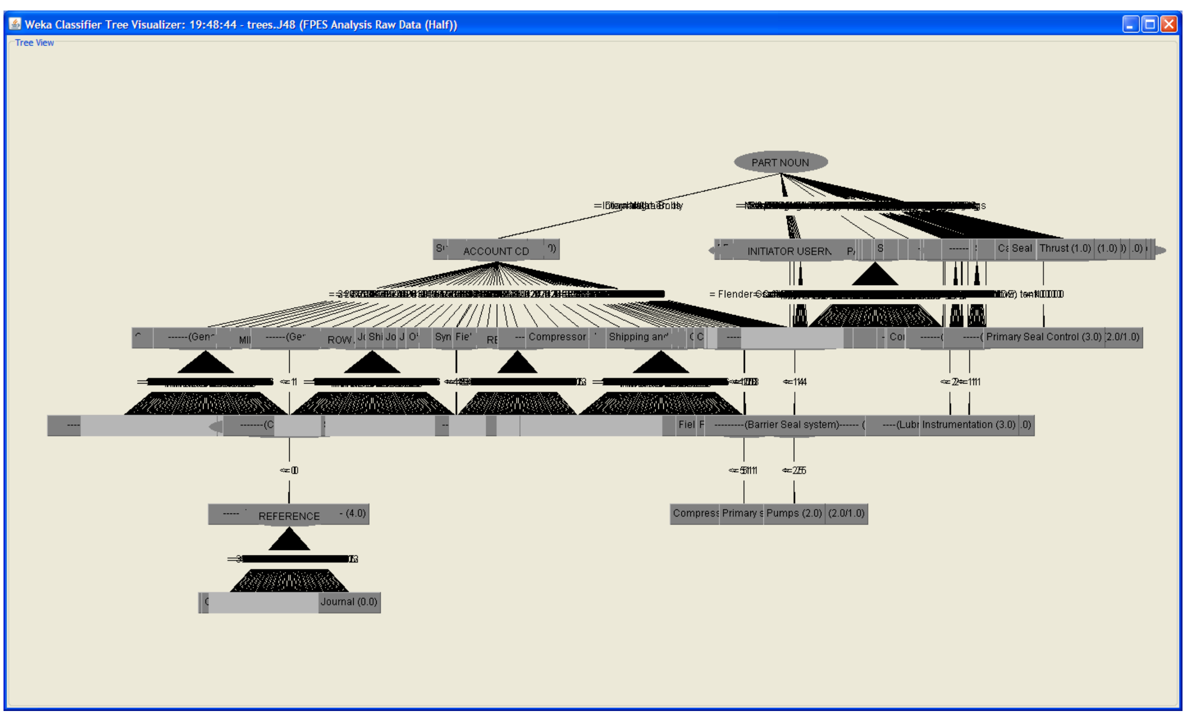

Figure A-6. Decision Tree using Field Data

8. REFERENCES

André, S., & Paulo Augusto Cauchick, M. (2008). Failure mode and effects analysis (FMEA) in the context of risk management in new product development. *The International Journal of Quality & Reliability Management, 25*(9), 899.

Arvinder, P. S. L. (1998). Evolution of product warranty: a chronological study. *Journal of Management History, 4*(2), 124.

Bailey, D., Frank-Schultz, E., Lindeque, P., & Temple, J., III. (2008). Three reliability engineering techniques and their application to evaluating the availability of IT systems: An introduction. *IBM Systems Journal, 47*(4), 577.

Berry, M. J. A., Linoff, G., & ebrary Inc. (2004). Data mining techniques

for marketing, sales, and customer relationship managementpp. xxv, 643 p.). Available from http://site.ebrary.com/lib/unioncollege/Doc?id=10114278

Blischke, W. R., & Murthy, D. N. P. (1992). Product warranty management, A Taxonomy for Warranty Policies. *European journal of operational research, 62*(2), 127-148.

Campbell, G. (2006). Guidance for the Use of Bayesian Statistics in Medical Device Clinical Trials: Food and Drug Administration.

Esterman, M., Gerst, P., Stiebitz, P. H., & Ishii, K. (2005). A Framework for Warranty Prediction During Product Development. *ASME Conference Proceedings, 2005*(42150), 381-388.

Gough, W. S., Riley, J., & Koren, J. M. (1990). *New approach to the analysis of reliability block diagrams.* Paper presented at the 1990 Proceedings - Annual Reliability and Maintainability Symposium, January 23, 1990 - January 25, 1990, Los Angeles, CA, USA.

Javier, P., Raul, P., Paolo, P., & David de la, F. (2002). A decision support system for applying failure mode and effects analysis. *The International Journal of Quality & Reliability Management, 19*(2/3), 137.

Jin, G., & Thomson, V. (2003). A new framework for MRP systems to be effective in engineered-to-order environments. *Robotics and Computer-Integrated Manufacturing, 19*(6), 533-541.

József, V., & Gábor, H. (2008). Selection methods for extended least squares support vector machines. *International Journal of Intelligent Computing and Cybernetics, 1*(1), 69.

Kim, H., & Sohn, S. (2008). Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research, 201*(3), 838.

Kmenta, S., & Ishii, K. (2004). Scenario-based failure modes and effects analysis using expected cost. *Transactions of the ASME. Journal of Mechanical Design, 126*(Copyright 2005, IEE), 1027-1035.

Lee, B. H. (2002). *Failure modes and effects analysis with Bayesian belief networks: bridging the design-diagnosis modeling gap.* Unpublished Thesis (Ph D), Stanford University, 2002.

Matic, Z., & Sruk, V. (2008). *The Physics-of-Failure approach in reliability engineering.* Paper presented at the ITI 2008 30th International Conference on Information Technology Interfaces, June 23, 2008 - June 26, 2008, Cavtat/Dubrovnik, Croatia.

Mazzuchi, T. A., & Soyer, R. (1993). A Bayes method for assessing product-reliability during development testing. *Reliability, IEEE Transactions on, 42*(3), 503-510.

Mueller, R. H. (2007). Design for Warranty Cost Reduction.

Romanowski, C. J. (2004). *A data mining-based engineering design support system.* State University of New York at Buffalo, United States -- New York.

Simpson, T. (2004). Product platform design and customization: Status and promise. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 3-20.

Sinha, S. (2007). CS498-SS Algorithms in Bioinformatics. from http://www.cs.uiuc.edu/class/fa07/cs498ss/lectures/Lecture14.pdf
Stamatis, D. H. (2003). *Failure mode and effect analysis : FMEA from theory to execution* (Vol. 2nd ed., rev. and expanded). Milwaukee, Wisc. :: ASQ Quality Press,.
Wilson, E. (1993). Maximizing Designers' Impact on Market Success through Product Definition: Design Management Journal.
Witten, I. H., & Frank, E. (2005). *Data mining : practical machine learning tools and techniques* (2nd ed.). Amsterdam ; Boston, MA: Morgan Kaufman.
Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1.
Yadav, & Prakash, O. (2002). An integrated framework for reliability prediction during product development process: Wayne State University.