PROJECT EVALUATION USING FUZZY LOGIC AND RISK ANALYSIS TECHNIQUES

by

Nabil D. Parsiani Shull

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
INDUSTRIAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2006

Approved by:

_____	_____
William Hernández-Rivera, PhD Member, Graduate Committee	Date
_____	_____
Rogelio Palomera García, PhD Member, Graduate Committee	Date
_____	_____
Viviana Cesaní Vázquez, PhD President, Graduate Committee	Date
_____	_____
Raúl Macchiavelli, PhD Representative of Graduate Studies	Date
_____	_____
Agustín Rullán Toro, PhD Chairperson of the Department	Date

ABSTRACT

The purpose of this thesis work is to develop a new methodology to solve risk analysis problems with the purpose of determining the project's attractiveness. The algorithm created in this thesis was developed using fuzzy logic and designed for the software development industry. Fuzzy logic was used since it is a tool capable of modeling complex and uncertain or vague data using simple terminology such as IF-Then statements. This logic is perfect to deal with the uncertainty risk plays in a projects development.

This methodology provides a quick and efficient tool for project managers in their use of project evaluation, by allowing the project manager to scrap useless projects without putting the least amount of effort into an analysis. The result of this work is the development of a new line of thought in the area of risk analysis in software development, where the probability and impact of a project delay can determine the attractiveness of the project. Yet, this methodology can also be generalized and therefore have the capability of being used in the project evaluation in many different kinds of industries not only the software industry.

RESUMEN

El propósito de este trabajo de tesis es desarrollar una nueva metodología para resolver problemas de análisis de riesgo con el propósito de determinar cuán atractivo es un proyecto. El algoritmo creado fue desarrollado usando lógica difusa y está diseñado para la industria de desarrollo de programas de computadora. Lógica difusa fue usada ya que es una herramienta capaz de modelar datos complejos y que sufren de incertidumbre, usando una terminología tan simple como los incisos de lógica "Si – Entonces". Esta lógica es apropiada para atender la incertidumbre inherente que presenta la evaluación de riesgo en el desarrollo de proyectos.

Esta metodología provee una herramienta rápida y eficiente en la evaluación de proyectos, al permitir que se eliminen proyectos no atractivos sin gastar el mínimo esfuerzo en análisis. El resultado de este trabajo es el desarrollo de una nueva línea de pensamiento en el área de análisis de riesgo en el desarrollo de programas, donde la probabilidad e impacto del atraso pueda determinar cuán atractivo sea el proyecto.

DEDICATION

To my loved ones,

I want thank you all for always being there for me through thick and thin, always supporting me and encouraging me. I appreciate now more than ever all of the things you have all done for me. I especially want to thank my Mother, my Father, my Sisters, and my Girlfriend, for their encouragement and love.

ACKNOWLEDGEMENTS

First I would like to thank the University of Puerto Rico for giving me the opportunity to study my master degree here. My deepest thanks to my thesis advisor, Dr. Viviana Cesani; her help and support in the development of this thesis will always be greatly appreciated and I thank her for her friendship. I would like to give my appreciation to committee members Dr. Rogelio Palomera, for his aid in the development of the technical aspect of this thesis and his long time friendship, and Dr. William Hernandez-Rivera for his support, understanding, and flexibility.

I would like to give a special thanks to my father, Hamed, for providing me with this excellent education, my mother Virginia, for always helping me out even if it was at 3:00am, my older sister, Monireh, for always providing me with sound counsel, and my younger sister, Tahireh, for her love and support.

Last, but certainly not least I would like to thank Beatriz Cruz, since she has been my inspiration, the one person who has kept me working hard all these years and who has also been my best friend.

TABLE OF CONTENTS

TABLE LIST

Tables **Page**

FIGURE LIST

Figures **Page**

1 INTRODUCTION

Project managers all around the world have to make difficult decisions that could ultimately affect the stability and security of the company. The most difficult decision they must face is determining what projects to undertake and consequently invest money, time, and effort in them. By choosing to invest in a project it is very important that the project be fruitful or else the company ends up losing money, time, and valuable resources that could be used in a more useful endeavor. Consequently project managers have to take into consideration many factors before committing valuable resources to any project. These factors include, but are not limited to, time constraints, tangible costs, and profits. Most of the factors that adversely affect the project attractiveness are called risks, and generally risk is intangible and hard to measure.

Due to the uncertain nature of risk, project managers must somehow determine the impact the risks will have on the project. Good project managers are those that can determine the largest amount of risks and the impact these will have on the project. The impact that risk has on a project is quantified in terms of dollars; how much money would be lost. More often than not, Project Managers try to assess risk using exact values and fail. Since risk can not be quantified in straight, crisp terms it must be taken and analyzed as a distribution.

The software development industry is probably one of the most risky of industries at the present moment. Risk factors are present throughout the whole development

process and these can negatively affect the project. These software risk factors are of concern since there has been a large amount of software disasters occurring recently. “A recent survey of 600 firms indicated that 35 percent of them had at least one runaway software project.” [1]

1.1 DEFINITION OF THE PROBLEM

Every project that a software company is interested in undertaking includes some sort of risk. These risks can be detrimental to the company, and therefore they must be identified and assessed to determine the impact they may have on the company. Some projects will have more risks involved and probably have greater impacts on the company. It is important to determine the risks and their impact that a given project entails so as to determine the attractiveness of a project. For example, if a project with a high risk of failure whose negative impact will outweigh its benefits is an unattractive project and therefore should not be undertaken.

1.2 MOTIVATION

1.2.1 Justification

Various researchers have analyzed investments and projects by means of cash flows and some have made the effort to take into account the risk involved in these investments and projects. These researchers, though, have faced problems such as the effort, the cost, and the complexity that these analysis methods entail, causing them to be

fruitless. Also, in general many of these analysis methods are not very representative since many times the researchers are working with various variables that contain uncertainty. Thus, there is a need for a method or tool that simplifies the analysis for project selection and is able to deal with the problem of uncertainty.

Risk analysis has in its essence uncertainty and impreciseness. Any analysis made ignoring this uncertainty and impreciseness may cause information to be seriously misleading, therefore, contributing to large mistakes

Fuzzy logic is based upon uncertainties where there is an inherent impreciseness. It provides mathematical tools for solving and working out approximate reasoning processes when having to deal with imprecise, uncertain, and vague data. This logic is composed of fuzzy sets, which are provided by a mathematical definition rising from the concept of degrees of membership, which increases the number of possibilities that can be subject to research. .

Quoting a recent piece of literature which contributes immensely to the justification of this research, the author specifies under the area of future work that "the risk assessment process, guided by the process of causal mapping, introduces the concept of expressing the degree to which a risk factor exists in a project, and also the impact such a risk factor has on related risks. Stakeholders are commonly unable or unwilling to give precise values to these, but are usually more open to expressing them as an (expert) opinion, i.e. as a 'fuzzy' value that does not commit the participant to likely inaccuracy…

Fuzzy representation and reasoning approaches and neuro-fuzzy and system dynamics techniques may well hold the key to usefully capturing such risk data, and making the ensuing models functional". [1]

1.2.2 Objectives

The objectives of this research are centered on the development of an algorithm that is capable of solving a general representative problem to determine a project's attractiveness using fuzzy logic and based upon risk factors that plague the software industry. Other objectives include creating a robust risk classification and measurement system, and simplifying and reducing the effort and time it takes to perform an analysis for project selection with an investment risk analysis technique using fuzzy logic to solve for uncertainties.

1.2.3 Focus

The focus in this thesis work is centered upon determining the attractiveness of an investment project in the area of software development. Since real projects are limiting in their scope, a hypothetical project will be created in order to fully develop this research.

1.2.3 Thesis Organization

The remaining of this document is organized in the following way:

Chapter 2 is dedicated to the revision of the literature that provided important aspects of this thesis. The literature referred to in this section provided a key role in the development of this thesis, such as the case study used.

Chapter 3 is dedicated to the fundamentals which are the basis of the analysis in this thesis. These fundamentals include, what is fuzzy logic and how it works, and what is risk and everything it entails.

Chapter 4 is dedicated to the methodology undertaken in the development of the algorithm and models that compose this thesis.

Chapter 5 is dedicated to the actual development of the algorithm and models as well as the testing of the algorithm in various scenarios.

Chapter 6 is dedicated to the conclusions, recommendations, and the future work of this thesis.

2 LITERATURE REVIEW

Extensive research has been done to develop sophisticated tools that can analyze and provide accurate information for the choice of investments and projects. Many of these researchers though, have faced the dilemma that much of their data is plagued by uncertainty, vagueness and approximation. This uncertainty, vagueness, and approximation can be seen mostly in the area of risk analysis where risk is considered a black box in which the few who have ventured to take it into consideration and analyze it have failed to come back out of the box with a clear and understandable analysis, and therefore the majority of project managers have been afraid to even delve into this area. Those project managers that are successful are some of the few that have stuck their head into the box and these are either lucky or they have excellent foresight.

Since risk has no exact value, traditional quantitative risk assessments are usually qualified with a statement of uncertainty. Mahant (2004) 'presents a novel approach to overcome the fuzziness in traditional risk assessment, and create a risk assessment model using fuzzy logic'.

"The likelihood of occurrence of incidents is regarded as a function of the robustness of Safety Management System (SMS). Fuzzy logic is used to characterize the robustness of the SMS as the variable which determines the likelihood of incidents. Fuzzy logic is used to characterize consequences and then fuzzy set operations used to

combine the severity of consequences and likelihood of occurrence to calculate risk. The model assesses risk of one major hazard at a time."[2]

This work provides a good example and guide to processing vaguely defined variables, and variables whose relationships cannot be defined by mathematical relationships. It takes into account the vagueness and uncertainty inherent in risk and provides a good assessment based upon experts judgment. It also provides a guide to the construction of a fuzzy risk model, unfortunately though it does not relate directly with project evaluation, but rather with onsite safety risk assessment.

Many projects have been analyzed using risk, as mentioned earlier, but a research project performed by J.H.M Tah and V. Carr (2000) provides a very insightful use of fuzzy logic in a very complex project, full of risk. This research creates 'a hierarchical risk breakdown structure representation used to develop a formal model for qualitative risk assessment. It also provides a common language for describing risks, including terms for quantifying likelihoods and impacts so as to achieve consistent quantification. Where the relationships between risk factors, risks and their consequences are represented on cause and effect diagrams. And finally, a methodology for evaluating the risk exposure, considering the consequences in terms of time, cost, quality, and safety performance measures of a project based on fuzzy estimates of the risk components.'

This research provides a very detailed decomposition and analysis of risk, taking into consideration uncertainty. Yet, the focus is not in project evaluation, but rather its focus is only in the assessment of risk.

Since this thesis work is focusing on project evaluation in the area of software development it is very important to determine the risks that plague this industry. A publication presented by the Software Engineering Institute (1993) provides a very robust list of risks as a basic Work Breakdown Structure. This list is a taxonomy of risks divided into families, groups, and subgroups which gives an extremely broad description of the risks that plague the software development industry. This paper also provides a questionnaire that can be used by project managers to assess the risks that they encounter.

However, for a more practical and visual approach the paper published by (Al-Shehab et. al, 2005) provides a cause and effect diagram which allows project managers to have a better comprehension of the effects risk have upon a project. This paper also states the need for an analysis that can take into consideration the uncertainty that plagues the software development industry since crisp values can not be given to risks. It actually states that "fuzzy representation and reasoning approaches… may well hold the key to usefully capturing such risk data, and making the ensuing models functional."

All of these research projects performed in the past provide a very useful guide to individual aspects of this thesis work, which will incorporate many tools that have previously been presented.

3 THEORETICAL BACKGROUND

Part of this Theoretical Background was extracted from the previous work of José De Jesús [10] since he provides a detailed account of the history, mathematics, and basics of fuzzy logic. The second part of this Theoretical Background is the introduction of what is risk and what is the role of risk management in software development.

3.1 HISTORY OF FUZZY LOGIC

The history of fuzzy logic is quite interesting since many researchers delve deep into the past to determine the evolution of this type of logic. Some even go all the way back to approximately 300BC, where Aristotle, a Greek philosopher and scholar, together with preceding philosophers "devised a concise theory of logic and mathematics, the so-called Laws of Thought." [12]

A specific law of these "Laws of Thought" that is quite interesting is the "Law of the Excluded Middle" which states that every proposition must be either True or False [13]. Heraclitus contradicted this law proposing that things could be simultaneously True and not True.

The scholar and philosopher indicated that there is a third region between True and False where the opposites tumble about. This might as well have laid the foundation for what would become fuzzy logic.

Also interestingly the great mathematician and philosopher Bernard Russell found a very big gap in the classical set theory which is nothing but an old and well known ancient Greek paradox "a Grecian says that all the Grecian are liars." Is this man lying? If he is lying then he is saying the truth about Grecians, and therefore he is not lying. If he is saying the truth then he is lying since all Grecians are supposedly liars. Both situations are conflicting in essence since the clause is both true and false at the same time. From the point of view of classical logic, this is absolutely a paradox and hence there is no solution. Later Lukasiewicz, a polish mathematician, proposed a systematic alternative to the bi-valued logic of Aristotle, and in the 1920s he described a three-valued logic, where the third value he proposed can best be translated as the term "possible," and he assigned it a numeric value between True and False [13]. Eventually, he proposed an entire notation and an axiomatic system from which he hoped to derive modern mathematics.

Later, he explored four-valued and five-valued logics, and then declared that in principle there was nothing to prevent the derivation of an infinite-valued logic. Lukasiewicz felt that three- and infinite-valued logics were the most intriguing, but he ultimately settled on a four-valued logic because it seemed to be the most easily adaptable to Aristotelian logic [13].

Knuth proposed a three-valued logic, similar to Lukasiewicz's, from which he speculated that mathematics would become even more elegant than in traditional two-value logic [17]. His insight, apparently missed by Lukasiewicz, was to use the integral

values [-1, 0, +1] rather than [0, 1, 2]. Nonetheless, this alternative failed to gain acceptance and passed into relative obscurity.

The notion of fuzzy set appears for the first time in a memo from the University of California at Berkeley written by Lotfi A. Zadeh in 1964. It was later published in The Information and Control Journal [20]. From that time on, it has served as the foundation for numerous papers on fuzzy sets from authors all over the world and set the basis for fuzzy logic. Fuzzy logic is basically a logic with multiple values, which allows values between the conventional evaluations of the precise logic 1 and 0. It also includes operations for 'and', 'or', 'not' and 'if-then'.

Hence, fuzzy logic extends conventional Boolean logic to handle the concept of the partial truth – the values falling between "totally true" and "totally false". These values are dealt with using degree of membership of an element to a set. The degree of membership can take any real value in the interval [0, 1]. Fuzzy logic makes it possible to imitate the behavior of human logic, which tends to work with "fuzzy" concepts of truth.

Although fuzzy control was not the first engineering application of fuzzy logic, it was the first application that drew huge attention to the practical potential of fuzzy set theory [3]. It uses many elements of fuzzy logic to define a rule-base for the controller.

3.2 ELEMENTS OF FUZZY SETS

3.2.1 Basic Concepts

In classical or crisp sets, an element in the universe has a well defined membership or non membership to a given set. Membership to a crisp set F can be defined through a membership function defined for every element *x* of the universe as

$$\mu_F(x) = \begin{cases} 1 & x \in F \\ 0 & x \notin F \end{cases} \tag{3.1}$$

An example of a graphic for the membership function of a crisp set is illustrated in Figure 1. Here, scanning the universe, there is an abrupt and well-defined transition from membership to non membership and vice versa. It is said to be "crisp".

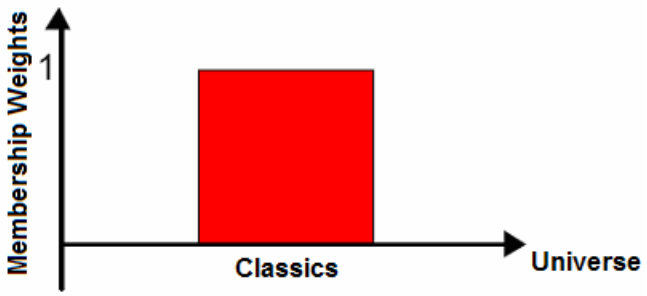

Figure 1: Classic Sets

For an element in a universe with fuzzy sets, the membership transition can be gradual. So the membership function can take any value between 0 and 1. This transition among various degrees of membership can be thought of as conforming to the fact that the boundaries of the fuzzy sets are vague and ambiguous. Fuzzy membership counterpart for Figure 1 would be that of Figure 2. Hence, membership of an element

from the universe in this set is measured by a function that attempts to describe vagueness and ambiguity. In fuzzy logic, linguistic variables take on linguistic values which are words (linguistic terms) with associated degrees of membership in the set. Thus, instead of a variable height assuming a numerical value of 1.75 meters, it is treated as a linguistic variable that may assume, for example, linguistic values of “tall” with a degree of membership of 0.92, "very short" with a degree of 0.06, or "very tall" with a degree of 0.7. Each linguistic term is associated with a fuzzy set, each of which has a defined membership function (MF).

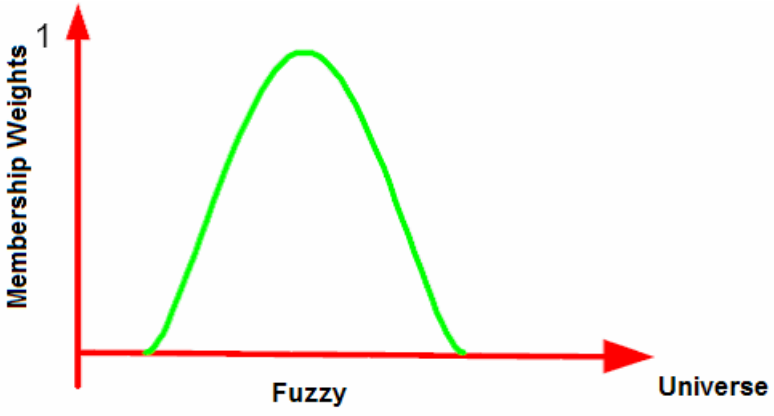

Figure 2: Fuzzy Sets

Formally, a fuzzy set is defined as a set of pairs where each element in the universe U has a degree of membership associated with it:

$$F = \{(x, \mu_F(x)) \mid x \in U, \mu_F(x) \in [0, 1]\} \quad (3.2)$$

$\mu_F(x)$ is known as the membership function of the set F. Most often, one refers to the fuzzy set just by mentioning the membership function, the universe being implicit.

The value $\mu_F(x)$ is the degree of membership of object x to the fuzzy set F where $\mu_F(x) = 0$ means that x does not belong at all to the set, while $\mu_F(x) = 1$ means that the element is totally within the set [3].

As an example consider the ambient temperature with the concepts of hot and cold. The membership functions in the classical logic for this example are given by

$$\mu_{Hot}(x) = \begin{cases} 1 & \text{if } x \geq 30^{\circ}C \\ 0 & \text{if } x < 30^{\circ}C \end{cases} \tag{3.3}$$

and

$$\mu_{Cold}(x) = 1 - \mu_{Hot}(x) \begin{cases} 1 & \text{if } x \leq 30^{\circ}C \\ 0 & \text{if } x > 30^{\circ}C \end{cases} \tag{3.4}$$

The 30°C boundary in this example is arbitrary. Independently of this boundary value, classical logic cannot interpret intermediate values. A graph of the membership function for the classic temperature variable is shown in Figure 3.

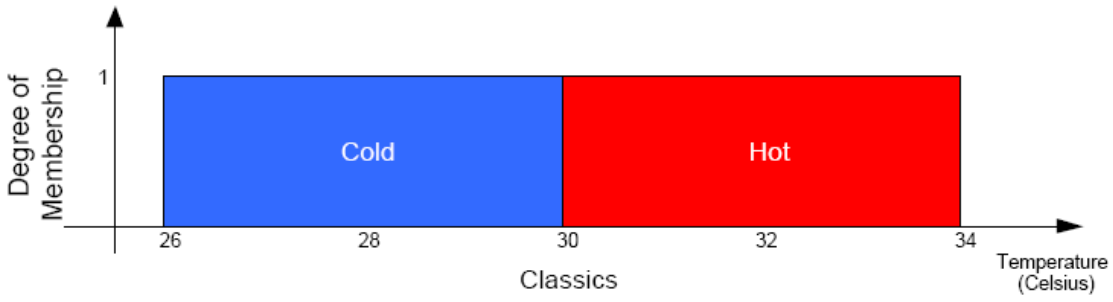

Figure 3: Classic Temperature Set

On the other hand, fuzzy logic solves the crisp problem with membership functions such as given by

$$\mu_{\text{Hot}}(x) = \begin{cases} 0 & \text{if } x \leq 29^{\circ}C \\ \frac{x-29}{2} & \text{if } 29^{\circ}C < x < 31^{\circ}C \\ 1 & \text{if } x \geq 31^{\circ}C \end{cases} \tag{3.5}$$

And

$$\mu_{\text{Cold}}(x) = 1 - \mu_{\text{Hot}}(x) = \begin{cases} 0 & \text{if } x \geq 31^{\circ}C \\ \frac{31-x}{2} & \text{if } 29^{\circ}C < x < 31^{\circ}C \\ 1 & \text{if } x \leq 29^{\circ}C \end{cases} \tag{3.6}$$

Figure 4 shows a representation of a fuzzy set using these membership functions. In this set, a 30.5°C temperature belongs 25% to cold and 75% to hot.

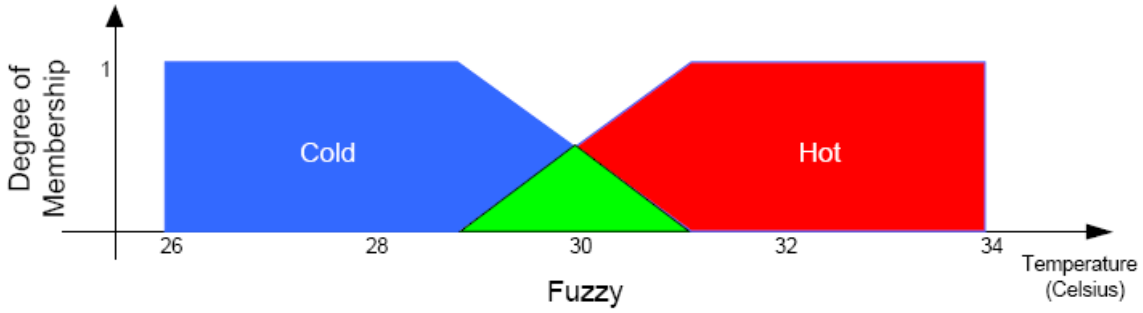

Figure 4: Fuzzy Temperature Set

Further refinements, such as introducing an intermediate "warm" temperature, can be made. A graph of the membership functions for the classic and fuzzy temperature sets are shown in Figure 5. In this Figure, temperature values between 29 and 31°C are

represented by other classic and fuzzy sets. Values from 29 to 30°C belong, with different extent, both to cold and warm membership functions. Similarly, values from 30 to 31°C belong, with different extent, both to warm and hot membership functions.

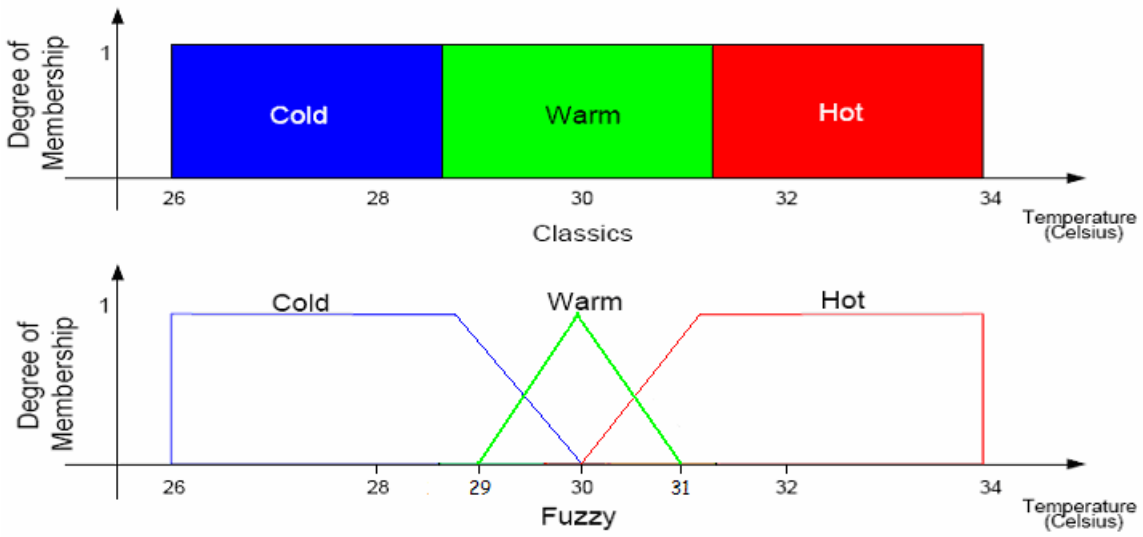

Figure 5: Fuzzy Temperature Set with more Precision

In any case, the crisp nature is always present in classical sets, while fuzzy sets allow gradual membership to better adapt to our subjective criteria. The membership function describes the degree of membership of the different elements of the fuzzy set in the universe. The selection of the form of membership function is subjective and depends on the context. However, for practical reasons, triangular, trapezoidal and bell shape functions are the most commonly used in engineering applications. These are shown in Figure 6.

3.2.2 Basic Operations on Fuzzy Sets

Fuzzy Sets can be operated with each other in the same way as classical sets.

Classical sets will be denoted by simple capital letters, like A, B, C, whereas fuzzy sets will be denoted by capital letters with a tilde underneath, i.e. $\underset{\sim}{A}$, $\underset{\sim}{B}$, $\underset{\sim}{C}$.

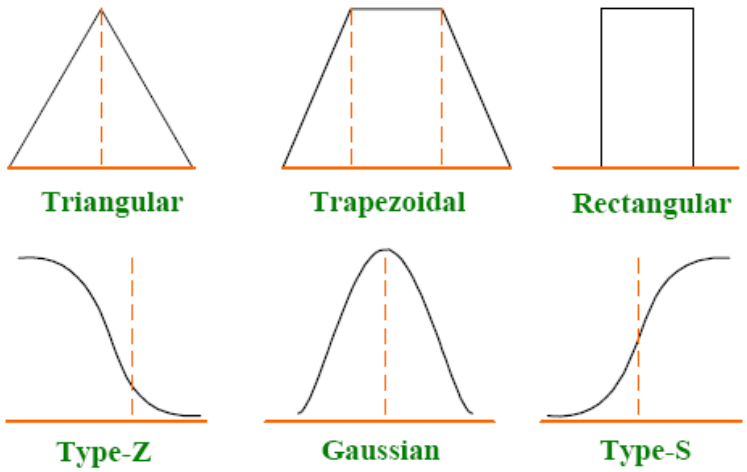

Figure 6: Different Types of Fuzzy Set Membership Functions

Empty and Universal Fuzzy Sets

A fuzzy set is empty if $\mu_{\emptyset}(x) = 0$ and universal if $\mu_x(x) = 1 \ \forall \ x \in U$.

Equal Sets

Two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are equal if

$$\mu_{\underset{\sim}{A}}(x) = \mu_{\underset{\sim}{B}}(x) \quad \forall \ x \in U. \tag{3.7}$$

Absolute and Relative Complements

The absolute complement (NOT) of a fuzzy set $\underset{\sim}{A}$ is denoted as

$$\mu_{\overline{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{B}}(x) \quad \forall \ x \in U. \tag{3.8}$$

Figure 7 shows absolute complements of a fuzzy set.

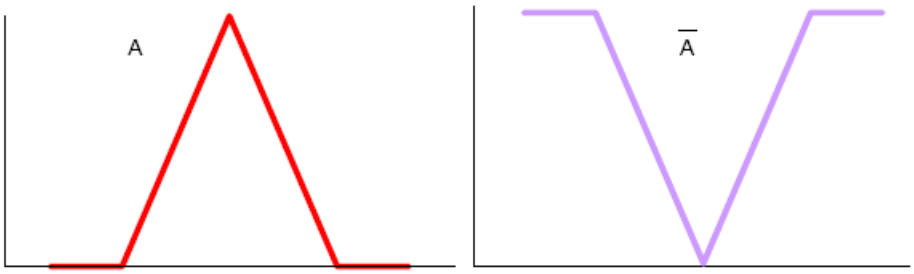

Figure 7: The Absolute Complement of a Fuzzy Set

The relative complement of $\underset{\sim}{A}$ with respect to $\underset{\sim}{B}$, denoted by $\underset{\sim}{B} - \underset{\sim}{A}$, is defined by

$$\mu_{\underset{\sim}{B}-\underset{\sim}{A}}(x) = \mu_{\underset{\sim}{B}}(x) - \mu_{\underset{\sim}{A}}(x) \tag{3.9}$$

provided that

$$\mu_{\underset{\sim}{B}}(x) \geq \mu_{\underset{\sim}{A}}(x) \tag{3.10}$$

Figure 8 shows relative complements of a fuzzy set.

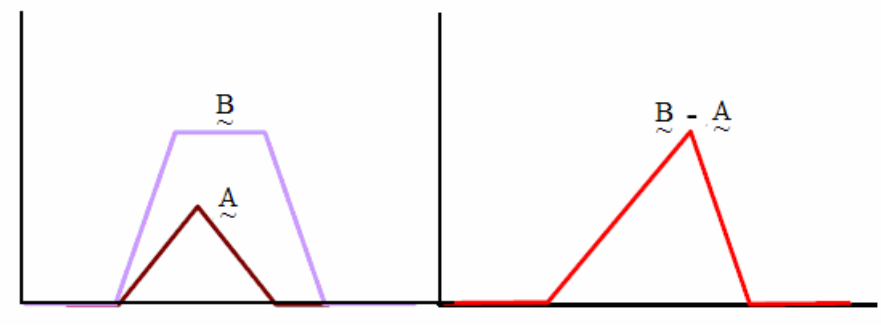

Figure 8: Relative Complement of a Fuzzy Set

The Union of fuzzy sets

The Union of Fuzzy Sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is a fuzzy set whose membership function for an element of the universe is the greatest or maximum of the membership functions of $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

$$\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x) = \max\left(\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right) \tag{3.11}$$

Figure 9 shows union of fuzzy sets.

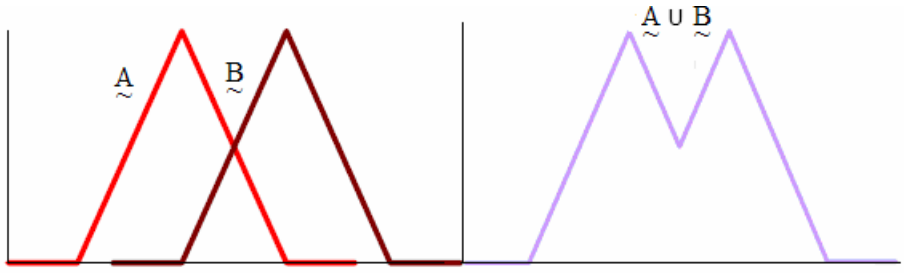

Figure 9: The Union on Fuzzy Sets

The Intersection of fuzzy sets

The intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is a fuzzy set whose membership function for an element of the universe is the lowest or minimum of the membership functions of $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

$$\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x) = \min\left(\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\right) \tag{3.12}$$

Figure 10 shows the intersection of fuzzy sets.

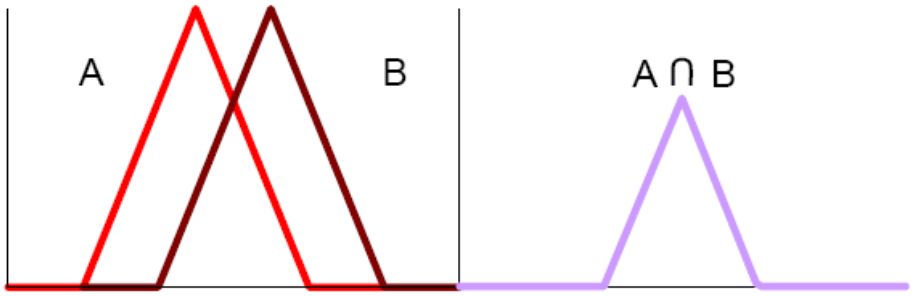

Figure 10: The Intersection of Fuzzy Sets

3.2.3 Crisp Logic

Logic can be a means to compel us to infer correct answers, but it cannot by itself be responsible for our creativity or for our ability to remember. In other words, Logic can assist us in organizing words to make clear sentences, but it cannot help us determine what sentences to use in various contexts. The interest in Logic arises from the study of truth in logical propositions; in classical predicate logic this truth is of binary nature: a proposition is either true or false [18].

Rules are a form of proposition. A proposition is an ordinary statement involving terms which have been defined. In traditional propositional logic, a proposition must be meaningful to call it “true” or “false,” whether or not we know which of these terms properly applies [16].

Logical reasoning is the process of combining given propositions into other propositions, and then doing this over and over again. A new proposition can be obtained

from a given one by prefixing the clause "it is false that… ". This is the operation of negation (denoted ~ p). Propositions can be combined in many ways, all of which are derived from three fundamental operations: Conjunction (denoted p $\wedge$ q), where we assert the simultaneous truth of two separate propositions p and q; Disjunction (denoted p $\vee$ q) where we assert the truth of either or both of two separate propositions; and, Implication (denoted p $\rightarrow$ q) which usually takes the form of an IF-THEN rule. The IF part of an implication is called the Antecedent, whereas, the THEN part is called the Consequent. Additionally, p $\leftrightarrow$ q is the equivalence relation; it means that p and q are either both true or both false.

A tautology is a proposition formed by combining other propositions (p, q, r, ...) which is valid regardless of the truth or falsehood of p, q, r, ... Tautologies are important because they represent valued reasoning. In traditional propositional logic there are two very important inference rules, *Modus Ponens* and *Modus Tollens*. These inference rules are tautologies and they work based on two premises:

Modus Ponens- Premise 1: “x is A”; Premise 2: “IF x is A THEN y is B”; Consequence; “y is B.” Modus Ponens is associated with the implication “A implies B” [A → B]. In terms of propositions p and q, Modus Ponens is expressed as $(p \wedge (p \rightarrow q)) \rightarrow q$ [16]. The rule is summarized as

$$\begin{array}{l} \underline{\text{Modus Ponens}} \\ \text{If x is A} \quad \text{then y is B} \\ \text{x is A} \\ \hline \therefore \text{ y is B} \end{array} \tag{3.13}$$

Modus Tollens- Premise 1: “IF x is A THEN y is B”; Premise 2: “y is not B”; Consequence: “x is not A.” in terms of propositions p and q, Modus Tollens is expressed as $(\sim q \wedge (p \rightarrow q)) \rightarrow \sim p$. [16]

$$\begin{array}{l} \underline{\text{Modus Tollens}} \\ \text{If x is A} \quad \text{then y is B} \\ \text{y is not B} \\ \hline \therefore \text{y is not A} \end{array} \tag{3.14}$$

Whereas, Modus Ponens plays a central role in engineering applications of logic, due in large part to cause and effect, Modus Tollens does not seem to have yet played much of a role. This could be due to the causal nature of the engineering applications.

3.2.4 Fuzzy Logic

Dr. Lotfi Zadeh (1965) from UC/Berkeley introduced Fuzzy Logic in the 60's as a means to model uncertainty in natural language [22]. Fuzzy logic extends conventional Boolean logic to handle the concept of the partial truth – with values really between "totally true" and "totally false" truths, making it easier to imitate the behavior of the human reasoning.

The extension of crisp logic to fuzzy logic is made by replacing the bivalent membership functions of crisp logic with fuzzy membership functions. Fuzzy values are assigned to evaluate the truth of propositions, and operations with these values are applied to evaluate composite propositions.

The meaning of a fuzzy proposition such as "x is A" is defined by the membership function that represents fuzzy set A. Assigning a meaning for compound fuzzy propositions, such as "x is A and y is B" or "(x is A or x is B) and y is C", implies the calculation of the membership function that characterizes the fuzzy relation induced by the proposition. In order to do so, it is necessary to define the interpretation for the linguistic connectives "and", "or" and for the operator "not".

The logical connectives of negation, disjunction, conjunction, and implication are also defined for fuzzy logic. These connectives are given in the followings equations for

two simple propositions: proposition $\underset{\sim}{P}$ defined on fuzzy set $\underset{\sim}{A}$ and proposition $\underset{\sim}{Q}$ defined on fuzzy set $\underset{\sim}{B}$. T ($\underset{\sim}{P}$) denotes the truth value of proposition $\underset{\sim}{P}$.

Basic connectives operations:

Negation

$$T(\overline{\underset{\sim}{P}}) = 1 - T(\underset{\sim}{P}) \tag{3.15}$$

Disjunction

$$\underset{\sim}{P} \vee \underset{\sim}{Q}: \text{x is } \underset{\sim}{A} \text{ or } \underset{\sim}{B} \qquad T(\underset{\sim}{P} \vee \underset{\sim}{Q}) = \max (T(\underset{\sim}{P}), T(\underset{\sim}{Q})) \tag{3.16}$$

Conjuntion

$$\underset{\sim}{P} \wedge \underset{\sim}{Q}: \text{x is } \underset{\sim}{A} \text{ and } \underset{\sim}{B} \qquad T(\underset{\sim}{P} \wedge \underset{\sim}{Q}) = \min (T(\underset{\sim}{P}), T(\underset{\sim}{Q})) \tag{3.17}$$

The main differences among the different inference schemes presented by different authors come from the interpretation of implication, that is, substituting the above mentioned operators by c-norms, s-norms, and t-norms, respectively. More about these operators can be found in [2].

Implication

$$\underset{\sim}{P} \rightarrow \underset{\sim}{Q}: \text{if x is } \underset{\sim}{A}, \text{ then y is } \underset{\sim}{B} \tag{3.16}$$

The most common interpretations that have been adopted are:

Zadeh

$$T(\underset{\sim}{P} \rightarrow \underset{\sim}{Q}) = T(\overline{\underset{\sim}{P}} \vee \underset{\sim}{Q}) = \max\,(T(\overline{\underset{\sim}{P}}), T(\underset{\sim}{Q})) \tag{3.17}$$

Mamdani

$$T(\underset{\sim}{P} \rightarrow \underset{\sim}{Q}) = \min\,(T(\underset{\sim}{P}), T(\underset{\sim}{Q})) \tag{3.18}$$

Larsen

$$T(\underset{\sim}{P} \rightarrow \underset{\sim}{Q}) = T(\underset{\sim}{P}) \cdot T(\underset{\sim}{Q}) \tag{3.19}$$

Engineers usually prefer Mamdani's or Larsen's formulas, while Zadeh's interpretation, derived directly from classical logic, is used in social sciences.

Modus Ponens is a rule of inference pertaining to the *if/then* operator. Modus Ponens states that if the antecedent of a conditional is true, then the consequent must also be true. Modus Tollens also is a rule of inference pertaining to the *if/then* operator. Modus Tollens states that if the consequent of a conditional is false, then the antecedent must also be false.

Fuzzy inference refers to computational procedures used for evaluating fuzzy linguistic descriptions. There are two important inference procedures: Generalized Modus Ponens (GMP) and Generalized Modus Tollens (GMT).

In fuzzy logic, Modus Ponens is extended to form Generalized Modus Ponens- Premise 1:“IF u is A THEN v is B”; Premise 2: “ u is A' ”; Consequence: “ v is B' ”.

Compare Modus Ponens and Generalized Modus Ponens to see their subtle differences. Namely, in the latter, fuzzy set A' is not necessarily the same as rule antecedent fuzzy set A, and fuzzy set B' is not necessarily the same as rule consequent B.

GMP allows us to compute the consequent B'. It is formally stated as

Modus Ponens	Generalized Modus Ponens	
If x is A then y is B	If x is A then y is B	
x is A	x is A'	(3.21)
∴ y is B	∴ y is B'	

Consider a linguistic description involving only a simple *if/then* rule with known implication relation $\underset{\sim}{R}(x,y)$ and a fuzzy value A' approximately matching the antecedent of the rule. The consequent can be obtained by taking the composition of the fuzzy set A' and fuzzy relation $\underset{\sim}{R}(x,y)$, and in general is symbolically given as follows:

$$\mu_{B'}(y) = \bigvee_{x} \{ \mu_{A'}(x) \wedge \mu_{\underset{\sim}{R}}(x, y) \} \tag{3.22}$$

In GMT a rule and a fuzzy value approximately matching its consequent are given and it is desired to infer its antecedent

Modus Tollens	Generalized Modus Tollens	
If x is A then y is B	If x is A then y is B	
y is B	y is B'	(3.23)
∴ x is A	∴ x is A'	

Since Modus Tollens is seldom used, no further insight is needed for the purpose of this work.

3.3 FUZZY LOGIC SYSTEMS

A FLS receives a crisp input and may deliver either a fuzzy set or a crisp value. The basic FLS contains four components: a rule set, a fuzzifier, an inference engine, and a defuzzifier. Rules may be provided by experts or can be extracted from numerical data. In either case, the engineering rules are expressed as a collection of IF-THEN statements. These statements are related to fuzzy sets associated with linguistic variables [16].

The fuzzifier maps the input crisp numbers into the fuzzy sets to obtain degrees of membership. It is needed in order to activate rules, which are in terms of the linguistic variables. The inference engine of the FLS maps the antecedent fuzzy (IF part) sets into consequent fuzzy sets (THEN part). This engine handles the way in which the rules are combined. In practice, only a very small number of rules are actually used in engineering applications of Fuzzy Logic (FL) [8].

In most applications, crisp numbers must be obtained at the output of an FLS. The defuzzifier maps output fuzzy sets into a crisp number, which becomes the output of the FLS. In a control application, for example, such a number corresponds to a control action.

3.3.1 Fuzzification

The first step in fuzzy logic processing involves a domain transformation called fuzzification (Figure 11). Crisp inputs are transformed into fuzzy inputs. To transform crisp input into fuzzy input, membership functions must first be defined for each input.

Once membership functions are defined, fuzzification takes a real time input value, such as temperature, and compares it with the stored membership function information to produce fuzzy input values.

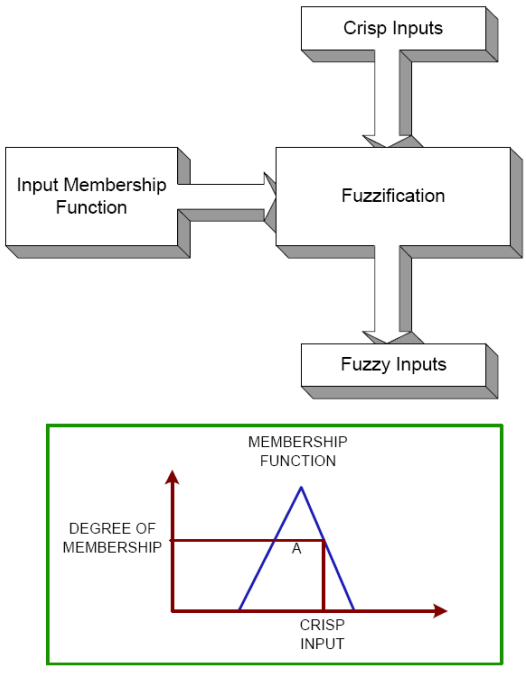

Figure 11: Fuzzification

The first step in fuzzification is to assign fuzzy labels in the Universe of discourse of each of the crisp inputs. So for temperature, we might assign a range of labels like those shown in Figure 12.

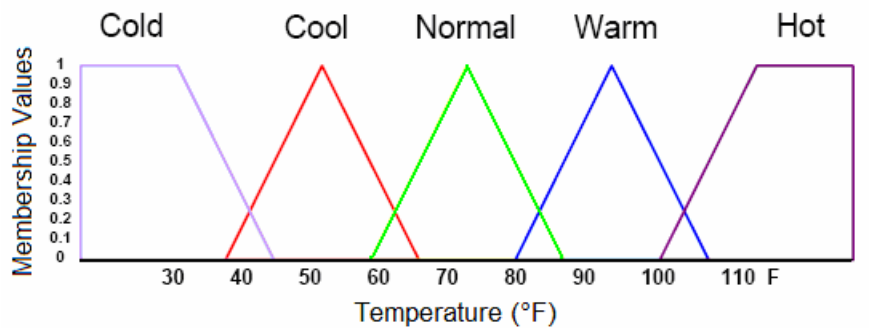

Figure 12: Temperature Membership Functions

Each crisp input into a fuzzy system can have multiple labels assigned to it. In general, the greater the number of labels assigned to describe an input variable, the higher the resolution of the resultant fuzzy control system, resulting in a smoother response.

However, a large number of labels require added computation time. Moreover, an excessive number of labels can lead to an unstable fuzzy system [3]. As a result, the most common number of labels for each variable in a fuzzy system falls between 3 and 9 [3]. The number is usually taken to be an odd number 3, 5, 7, 9, though this is not a requirement, but something that has been seen in applications [3].

The control surface fuzzy sets on each side of the zero (or normal) action set should be balanced and symmetric. Thus if you have a variable, Temperature, a fuzzy region LOW should also have a corresponding region HIGH as well as a normal temperature set of NORMAL.

Next, membership functions are defined to give numerical meaning to each label. Each membership function identifies the range of inputs values that corresponds to a label.

Unlike Boolean logic, the membership function of a label does not define boundaries where the label applies fully on one side of a cutoff and not at all on the other side of the cutoff. Instead there is a region where input values gradually change from being fully applicable to completely inapplicable.

Membership functions can have several different shapes, like those shown to the Figure 6. Trapezoidal, bell, and triangular are the most frequently used. Although other shapes may be more representative of natural occurring phenomena, they require more complicated equations or large look-up tables to be represented accurately.

A Fuzzy singleton is a fuzzy set whose support is a single point in U with a membership function of one. Singletons are easily represented in a computer and allow for simpler defuzzification algorithms. They are, therefore, frequently used to describe fuzzy outputs.

An input membership function is created by specifying a number, that is, the degree of membership, for each possible input value for a given label. The y-axis {μ} values refer to the degree to which the crisp input value (temperature) applies to each of the membership function labels (cool, warm, etc.). Input values can belong to more than one fuzzy set. Describing crisp inputs in fuzzy terms allows the system to gracefully respond to gradual change in input temperature.

3.3.2 Fuzzy Inference

Fuzzy logic based systems use RULES to represent the relationship between observations and actions. These rules consist of a precondition (IF-part) and a consequence (THEN-part). The precondition can consist of multiple conditions linked together with AND or OR conjunctions. Conditions may be negated with a NOT. The computation of fuzzy rules is called Fuzzy Inference.

Fuzzy rule inference consists of two steps:

- Inferencing, which determines the fuzzy subset of each output variable for each rule. Usually only MIN or PRODUCT are used as inference rules. In MIN inferencing, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth (fuzzy logic AND). In PRODUCT inferencing, the output membership function is scaled by the rule premises' computed degree of truth.

- Composition, which combines the fuzzy subsets for each output variable into a single fuzzy subset. Usually MAX or SUM are used. In MAX composition, the combined output fuzzy subset is constructed by taking the point wise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy logic OR). In SUM composition, the combined output fuzzy subset is constructed by taking the pointwise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.

IF-THEN rules are a common way of representing and communicating knowledge in everyday conversation. Anyone who has written a program or machine code knows how complicated (and difficult to debug, read, and maintain) the if-then lines can get. Fuzzy rules offer a way of getting around that by trading the precise representation of the values that variables must assume with much more intuitive fuzzy representations.

In binary logic the consequent is either true or false. In fuzzy logic partial truths are allowed so the consequent is as partially true as the antecedent allows it to be.

In general a rule by itself does not do much. What is needed are a set of rules that can play off one another. The fuzzy inference methodology allows “fair” competition between these rules to produce sophisticated answers using seemingly simple premises.

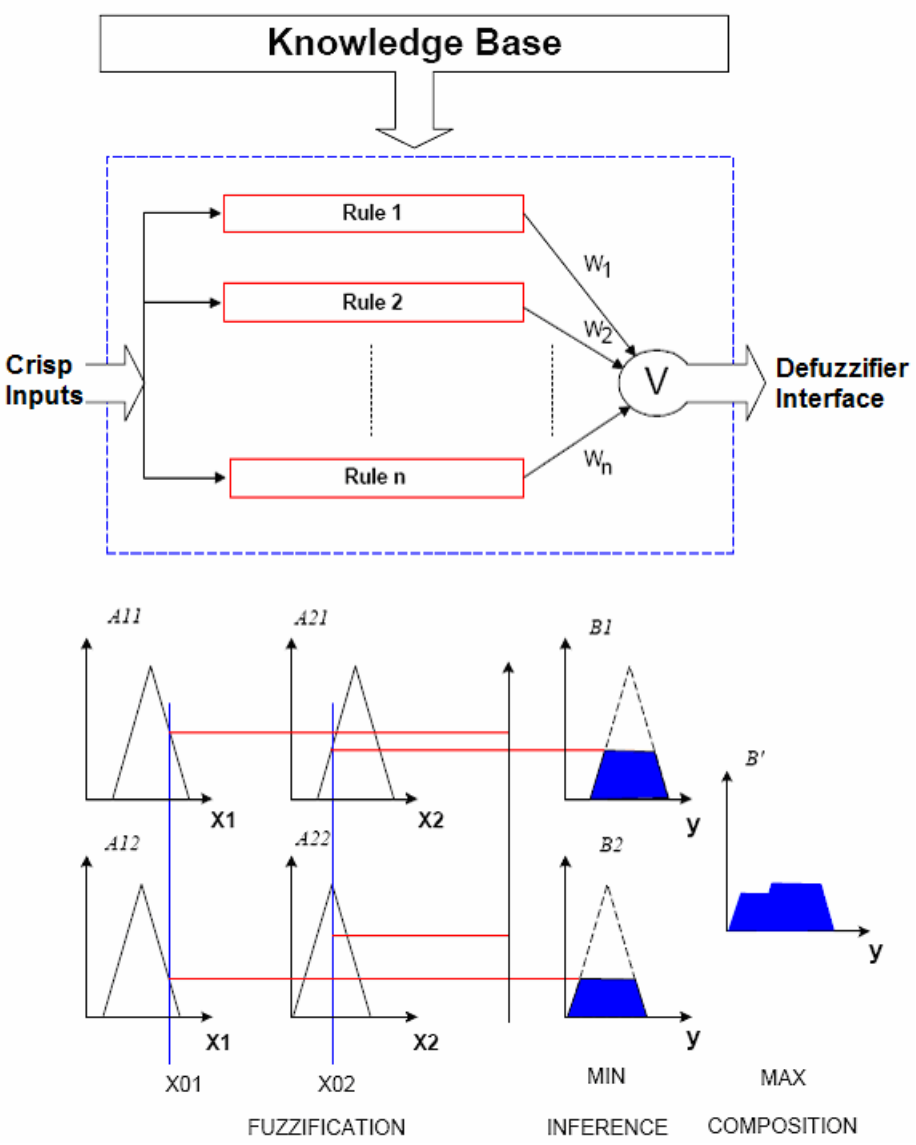

Figure 13: Inference Process

3.3.3 Defuzzification

This stage is used to convert the fuzzy output set to a crisp number. Two of the more common techniques are the Centroid and Maximum methods. In the Centroid method, the crisp value of the output variable is computed by finding the value of the center of gravity of the membership function. In the Maximum method, the crisp value of

the output variable is the maximum truth-value (membership weight) of the fuzzy subset. Figure 14 illustrates the complete process with an example.

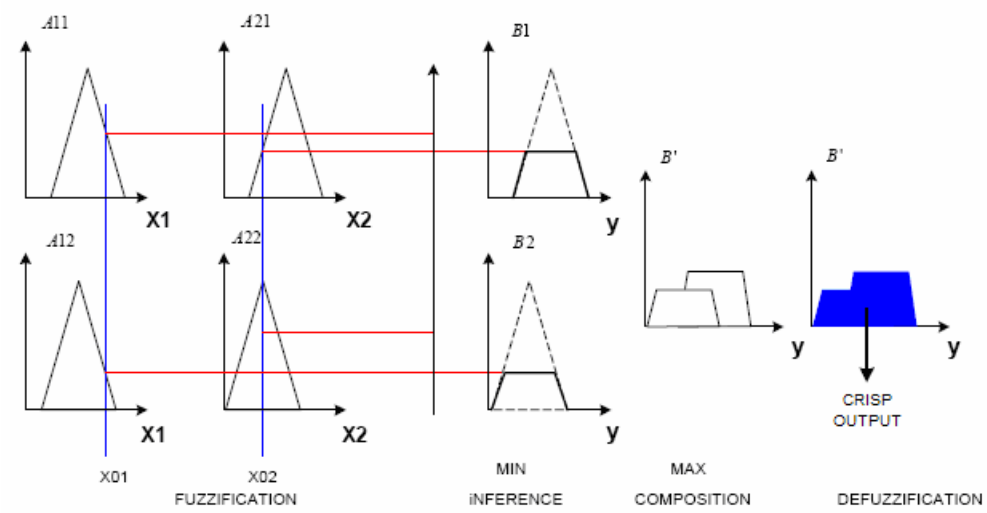

Figure 14: Fuzzy Logic System

Fuzzy logic is the tool that is used in this thesis yet the basis of this thesis lies in the concept of risk and risk management, specifically risk assessment and analysis in the area of software development. Following is a description of risk management, the definition of what is risk, and the roles that risk and risk management have in software development.

3.4 RISK MANAGEMENT

Risk Management is the area of project management that "identifies as many risk events as possible (what can go wrong), minimizes their impact (what can be done about the event before the project begins), manages responses to those events that do

materialize (contingency plans), and provides contingency funds to cover risk events that actually materialize."[7]

Another source defines risk management as "the act or practice of dealing with risk. It includes planning for risk, assessing (identifying and analyzing) risk issues, developing risk handling strategies and monitoring risks to determine how they have changed." [11].

Both of these definitions portray risk management as the process of planning for, identifying, analyzing, mitigating, and monitoring risks. The next section defines what is risk.

3.4.1 What is Risk?

'Risk', as defined by one source, "is the chance that an undesirable event will occur and the consequences of all its possible outcomes". [7]

A second source defines risk as "a measure of the probability and consequence of not achieving a defined project goal. Most people agree that risk involves the notion of uncertainty. However, when risk is considered, the consequences or damage associated with the event occurring must also be considered. Risk is not always easy to evaluate, since the probability of occurrence and the consequence of occurrence are usually not directly measurable parameters and must be estimated by statistical or other

procedures."[11] This is a perfect justification as to the development of the model proposed in this thesis, because fuzzy logic was built to deal with parameters that could only be estimated since exact values are impossible to determine.

"Risk has two primary components for a given event: probability (likelihood) of occurrence of that event and the impact of the event occurring (amount at stake)". [11]

Risk therefore can be conceptually defined as the function of likelihood and impact. Risk = *f* (likelihood, impact).[11] In the area of software development, the way they define the relation between likelihood and impact is that RE = P(UO) * L(UO), which translates as the Risk Exposure is equal to the Probability of an Unsatisfactory Output times the Loss caused by an Unsatisfactory Output.[4]

3.4.2 Components of Risk Management

As stated earlier the components of Risk Management are planning, assessment, handling, and monitoring. [11]:

- Risk planning is the process of developing and documenting an organized, comprehensive, and interactive strategy and methods for identifying and analyzing risk issues, developing risk handling plans, and monitoring how risks have changed.

- Risk assessment is the process of identifying and analyzing program areas and critical technical process risks to increase the likelihood of meeting cost, performance, and schedule objectives.
    - Risk identification is the process of examining the program areas and each critical technical process to identify and document the associated risk.
    - Risk analysis is the process of examining each identified risk issue to estimate the likelihood of a risk and predict the impact on the project.
- Risk handling is the process that identifies, evaluates, selects, and implements one or more strategies in order to set risk at acceptable levels given program constraints and objectives.
- Risk monitoring is the process that systematically tracks and evaluates the performance of risk handling actions against established metrics throughout the acquisition process and provides inputs to updating risk handling strategies, as appropriate.

3.4.3 Risk Management in Software Development

The software development industry is a very recent and upcoming industry, it is basically in its infancy and yet it is growing at alarming rates. This growth has also witnessed an incredible amount of project failures and difficulties, due to the changes that growth entails, such as software development techniques that are constantly evolving. This evolution entails a lack in the consolidation of a body of knowledge, which puts

pressure on companies to train and re-train their software engineers, as well as forcing norms and standards to constantly change.

As we can see this industry is extremely complex due to its flexible nature, the speed in which it is growing, and the risks it must face. The flexibility that it entails is the ability to change, manipulate, or even deliver incomplete programs into the market, causing an increase in competition and speed of delivery, yet sacrificing quality.

Since the software development industry has to deal with so many details and difficulties, it is important to be able to define all of these peculiar happenings with respect to the perspectives and goals of each individual stakeholder. This is why Risk Management has become an important aspect in the development of software. The software development industry has recently been delving into this area of Risk Management and therefore it is important to understand just what Risk Management means.

"There are many definitions of Risk Management that can be seen across the literature. Mcleod and Smith's (1996) definition suggests that project risk comprises chance encounters with events that may prevent the achievement of the project goal. Phillips (1998) dichotomizes Risk Management as 1) Risk Management is a set of tasks that address any potential problem in a project and 2) Risk Planning anticipates possible problems and appropriate actions. Schwalbe (2000) similarly suggests that Risk Management is a set of principles whereby the project manager continually assesses risks

and their consequences, and takes appropriate preventive strategies. McManus (2001) argued that project failure is caused by a combination of abnormal events or failures and the consequences those events have on the system."[9]

Henderson & Beaumont define Risk Management "as the sequence of actions occurring throughout the project life cycle and which the project manager continually assesses the potential negative effects of uncertainties and provides strategies and responses to minimize their effects on the project. Risk Assessment should occur throughout the project life cycle. The magnitude of the Risk Management task varies with the size of the project, and its importance." [9]

These definitions of Risk Management portray the role that a project manager must undertake to ensure the project results successful. The only problem is that very few project managers have the experience, knowledge, or even the desire to delve into the area of Risk Management. The software development industry has determined that probably the most important factor to the success of a project is the Management of Risk, and yet due to the complexity and difficulty of this task most project managers shy away, causing the company to take a leap of faith in the development of a project. This leap of faith opens up the project to a very high probability of failure. This lack of effort in the area of Risk Management due to its difficulty opens the doors to methods that can simplify the effort and time of the project manager, thus saving money and resources.

3.4.4 Risk Analysis

Risk analysis is the area of risk management that this thesis focuses on. As defined earlier risk analysis is the process of examining each identified risk issue to estimate the likelihood of a risk and predict the impact on the project.

The purpose in the development of a new risk analysis algorithm is to create a methodology capable of determining qualitatively the attractiveness of a project 'a priori', in other words, before investing any money or effort into the project. This new technique provides a quick and safe decision as to the security in undertaking a given project. If the decision is to undertake the project then other risk management procedures will be implemented.

3.4.5 Traditional Risk Analysis Techniques

Traditional risk analysis techniques have been used in the past to analyze the risks that a project may face in order to determine and quantify the impact in monitorial terms that the risks will have on a given project, and see whether one project is more attractive than another. Some of these techniques are probability trees, Monte Carlo simulation, decision trees, and expected value.

Probability tree diagrams describe the prospective cash flows, and the probability that each risk has of occurring. Decision tree diagrams depict and facilitate the analysis of problems that involve sequential decisions and variable outcomes over time. They are

also used to break down complex problems into smaller simple problems. Enable objective analysis and decision making that includes explicit consideration of the risk and effect of the future.

Monte Carlo simulation generates random outcomes for probabilistic factors so as to imitate the randomness inherent in the original problem. The first step is to construct an analytical model that represents the actual decision situation. The second step is to develop a probability distribution from subjective or historical data for each uncertain factor in the model. Sample outcomes are randomly generated by using the probability distribution for each uncertain quantity and are then used to determine a trial outcome for the model. Repeating this sampling process a large number of times leads to a frequency distribution of trial outcomes for a desired measure of merit. The resulting frequency distribution can then be used to make probabilistic statements about the original problem.

The expected value of a single random variable X, E(X), is a weighted average of the distributed values x that it takes on and is a measure of the central location of the distribution (central tendency of the random variable). The E(X) is the first moment of the random variable about the origin and is called the mean (central moment) of the distribution. [19]

The difficulty and failures of these techniques are that, with probability trees exact or discreet values of probability or impact can not be given to risk, while Monte

Carlo simulation requires that the analyst set up a mathematical model of the process. This setup can be very time consuming and the simulation may provide a very low benefit-cost ratio, and the problem with decision trees is that it requires the use of discrete attributes, and therefore, does not work well with uncertainty or vagueness. The expected value technique fails in that it forces the user to give a value of expected cost to that risk as well as an expected probability. The expected value technique fails in that it does not describe the distribution of these values around the mean, indicating that there might be large or small errors. In terms of this thesis these techniques can not be used as comparisons because these techniques provide a quantitative analysis while this thesis is a qualitative analysis.

This chapter was dedicated to the explanation of the tools used in this research and to the benefits this algorithm provides. The following chapter is dedicated to the methodology of this thesis.

4 METHODOLOGY

This thesis work required an intense research into the areas of risk analysis and software development. The problem that was determined and defined to be resolved was the result of realizing that the software development industry is plagued with risk, and risk analysis techniques are either too complex or they are inefficient since the software development industry continues to suffer from the effects of risk. The reason for this hole in the software development industry is due to the fact that many project managers are not willing to take into consideration risks because they are overwhelmed by the effort it takes, and also because the techniques that are existent force the project manager to give values that he or she really does not know. The fact that these techniques are very difficult and that they force the project manager to decide things that he or she is unsure about opens the doors for this thesis work since its' objective is to simplify the project manager's job and allow him or her to determine the risks that afflict the project by just using his or her expertise naturally (expressions) as opposed to forced (values).

The methodology undertaken to develop this investigation appears in Figure 15 and a brief description of each step is presented in the following sections.

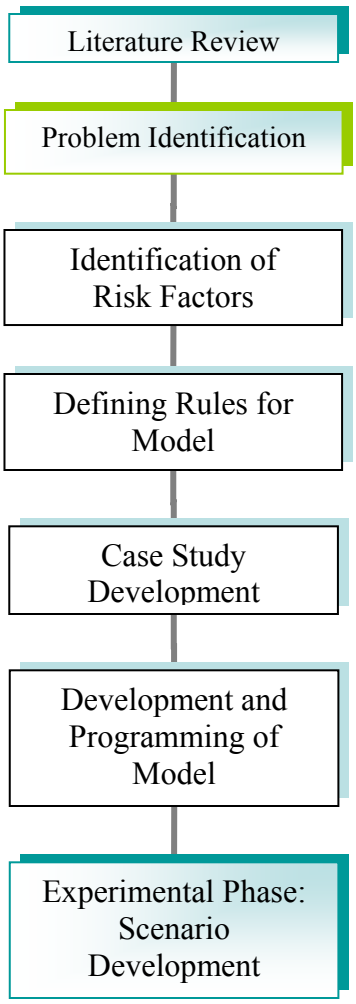

Figure 15: Methodology Used

4.1 LITERATURE REVIEW AND PROBLEM IDENTIFICATION

The first step in this research was a literature review in the areas of Project Evaluation, Software Development, and Risk Analysis. The result came to show that risk analysis is very important in the area of software development. The literature in the area of software development tends to consider risk and how to mitigate it upon the development of the project, but the literature is incomplete in the evaluation of a project's

attractiveness. Risk analysis has been researched quite deeply in the area of software development, that even a taxonomy of risk factors has been created to take into consideration all types of risk that may affect the development of any given project. Yet the interrelationships between these risks and the impact they will have upon the project have not been thoroughly researched. And, according to Al-Shehab and Hughes "Fuzzy representation and reasoning approaches techniques may well hold the key to usefully capturing such risk data, and making the ensuing models functional."[1]

4.2 IDENTIFICATION OF RISK FACTORS

The final objective of the proposed model is to provide an easy way of determining attractive projects for project managers. This can be done by identifying the relevant risks that each project entail and by determining their impact on the project via the natural spoken language, where the project manager does not need to know the exact (crisp) values.

Every project contains risks and these risks vary from project to project, yet in software development these are generally important risks that must be taken into consideration throughout the various stages of development of the software product.

These risks are compiled in a taxonomy for the software development industry. "This taxonomy might be used to classify many different factors associated with the development of software-dependent systems such as development tasks, quality

procedures, or sources or consequences of risk. However, the definitions as presented here are designed to facilitate classification of the risks themselves, as associated with the development process."[5] By using this taxonomy and a taxonomy-based questionnaire, Al-Shehab, Hughes, and Winstanley (2005) developed a Casual and Cognitive Map containing risks pertinent to their case study and their interrelationships. These risks, as defined by Al-Shehab, Hughes, and Winstanley (2005) through their research, will be used in this thesis work.

4.3 DEFINING THE RULES FOR THE MODELS

Fuzzy logic requires the user to determine a set of rules in order to create the model needed to solve the defined problem. These rules are defined based upon the opinions of experts in the field who know how the risks relate to each other and how they affect a given project. These rules are the basis of which the model will be developed, they are the building blocks of the model, and they are the center of scrutiny in the software development industry. A general model to encompass all of the software development industry areas is beyond the scope of this thesis. The source for the creation of the rules for this model is the research of Al-Shehab, Hughes, Winstanley (2005) and other authors, who have provided their insight into this area of risk via practical research. The result of collecting and analyzing this information is a defined interrelationship between the risks, where the risks are given membership functions as well.

Membership functions are fuzzy sets that determine vague concepts and that admit the possibility of partial membership. A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.

4.4 CASE STUDY DEVELOPMENT

The data used in the development of the model correspond to the expert opinion found in the literature relating to the area of software development. The software development industry has determined a widespread array of risks that plague the industry which have been collected and labeled in what is called a taxonomy of risks. This taxonomy is shown in Figure 16 and, though very good, it is extremely general and it does not specify what are the effects of these risks on the project. A research project undertaken in this field "proposed an evaluation framework for identifying the causes of shortfalls in implemented information system projects, which was developed during a longitudinal case study of a problematic project. This framework is a method of presenting in a visual fashion the factors that have a bearing on project failure and their interrelationships, which allows the stake holders to use the diagram to collaborate in the creation of risk models that can simulate the propagation and evolution of risks throughout the project life cycle."[1]

This research demonstrates that risk in software development is composed of various layers intertwined, causing an analysis to be very complex in nature. This research also provides a more specific study into the area of risk, based on the taxonomy

presented earlier. The end product that is produced from this research is a Casual and Cognitive Map that could map the "complex interactions between concept variables in a project environment."[1] This map, created from the case study of this research, which was based on the taxonomy of risk factors and a taxonomy-based questionnaire is the stepping stone of this thesis since it provides much needed cause and effect relationships for the development of a Fuzzy Logic Model.

4.5 DEVELOPMENT AND PROGRAMMING OF THE MODELS

Once identified the objectives of the model, the risk factors that are of importance, and the interaction of these risk factors, it was possible to begin to define the rules that compose the models. These rules are the main components of the models; they were developed using the Casual and Cognitive Map mentioned earlier [1], by gathering a large amount of data across the span of literature that encompasses the software development industry, and by sorting out the logic from the vast ocean of literature. This logic described in the literature is very subjective and varies a lot from researcher to researcher and from area to area. Therefore, these models are composed of logic where most of the researchers could agree upon as well as through my own perspective.

The models that are developed in this work are called the Project Delay Probability model, the Project Delay Impact model, and the Project Attractiveness model. The first two models were created from the Casual and Cognitive Map [1] and the third model is composed of the other two models. These models compose the proposed

algorithm and were programmed using the MATLAB software, where Fuzzy Logic techniques were implemented. To understand these models it is important to define what the following terms mean:

Project Delay Probability – this term refers to the probability that the project will be delayed due to the inherent risks in the project contributing to a project failure. This term has a very close relationship to the probability of a project failure.

Project Delay Impact – this term refers to the negative impact that a project delay will have upon a given project. This term has a very close relationship to the impact of a project failure.

Project Attractiveness – this term refers to how desirable a project is based upon how high the probability of failure is and how high is the impact of this failure.

It is also important to state that project delay is the most common variable used in the area of project management that relates to project failure. [9]

4.6 EXPERIMENTAL PHASE: SCENARIO DEVELOPMENT

Once the algorithm was developed and each model created, it was important to test the algorithm. These tests were conducted by creating a series of scenarios. Many of these scenarios work on the basis of common knowledge and from these it is easy to

determine whether a project is attractive or not. The other scenarios created are those to demonstrate the ability of this algorithm to sift easily between the input data and return a tangible and easily understood result, from which the common project manager can make a good decision.

This chapter was dedicated to the procedures underwent in the development of this thesis beginning with the research into the literature and ending with the testing of the created algorithm. The following chapter is dedicated to showing how the algorithm was developed and what the results are.

5 MODEL DEVELOPMENT AND ANALYSIS

5.1 RISK IDENTIFICATION

As mentioned in the methodology, the risks that are taken into consideration were extracted from a taxonomy of risks developed for the software development industry. This taxonomy of risks is an extremely general account of the risk factors that affect the software industry and can be seen in Figure 16 as a General Hierarchical Risk Breakdown Structure.

But since this taxonomy is extremely general and it is not portrayed as a cause and effect, a more specific case study ensues, where all the risks are portrayed in a causal and cognitive map "as a means of making visible the perceptions of project stakeholders with regard to the causes of shortcomings in completed IS/IT development projects." [1] This causal and cognitive map is presented in Figure 17.

A. Product Engineering

1. Requirements
- a. Stability
- b. Completeness
- c. Clarity
- d. Validity
- e. Feasibility
- f. Precedent
- g. Scale

2. Design
- a. Functionality
- b. Difficulty
- c. Interfaces
- d. Performance
- e. Testability
- f. Hardware Constraints
- g. Non-Developmental Software

3. Code and Unit Test
- a. Feasibility
- b. Testing
- c. Coding/ Implementation

4. Integration and Test
- a. Environment
- b. Product
- c. System

5. Engineering Specialties
- a. Maintainability
- b. Reliability
- c. Safety
- d. Security
- e. Human Factors
- f. Specifications

B. Development Environment

1. Development Process
- a. Formality
- b. Suitability
- c. Process Control
- d. Familiarity
- e. Product Control

2. Development System
- a. Capacity
- b. Suitability
- c. Usability
- d. Familiarity
- e. Reliability
- f. System Support
- g. Deliverability

3. Management Process
- a. Planning
- b. Project Organization
- c. Management Experience
- d. Program Interfaces

4. Management Methods
- a. Monitoring
- b. Personnel Management
- c. Quality Assurance
- d. Configuration Management

5. Work Environment
- a. Quality Attitude
- b. Cooperation
- c. Communication
- d. Morale

C. Program Constraints

1. Resources
- a. Schedule
- b. Staff
- c. Budget
- d. Facilities

2. Contract
- a. Type of Contract
- b. Restrictions
- c. Dependencies

3. Program Interfaces
- a. Customer
- b. Associate Contractors
- c. Subcontractors
- d. Prime Contractor
- e. Corporate Management
- f. Vendors
- g. Politics

Figure 16: Taxonomy of Software Development Risks [5]

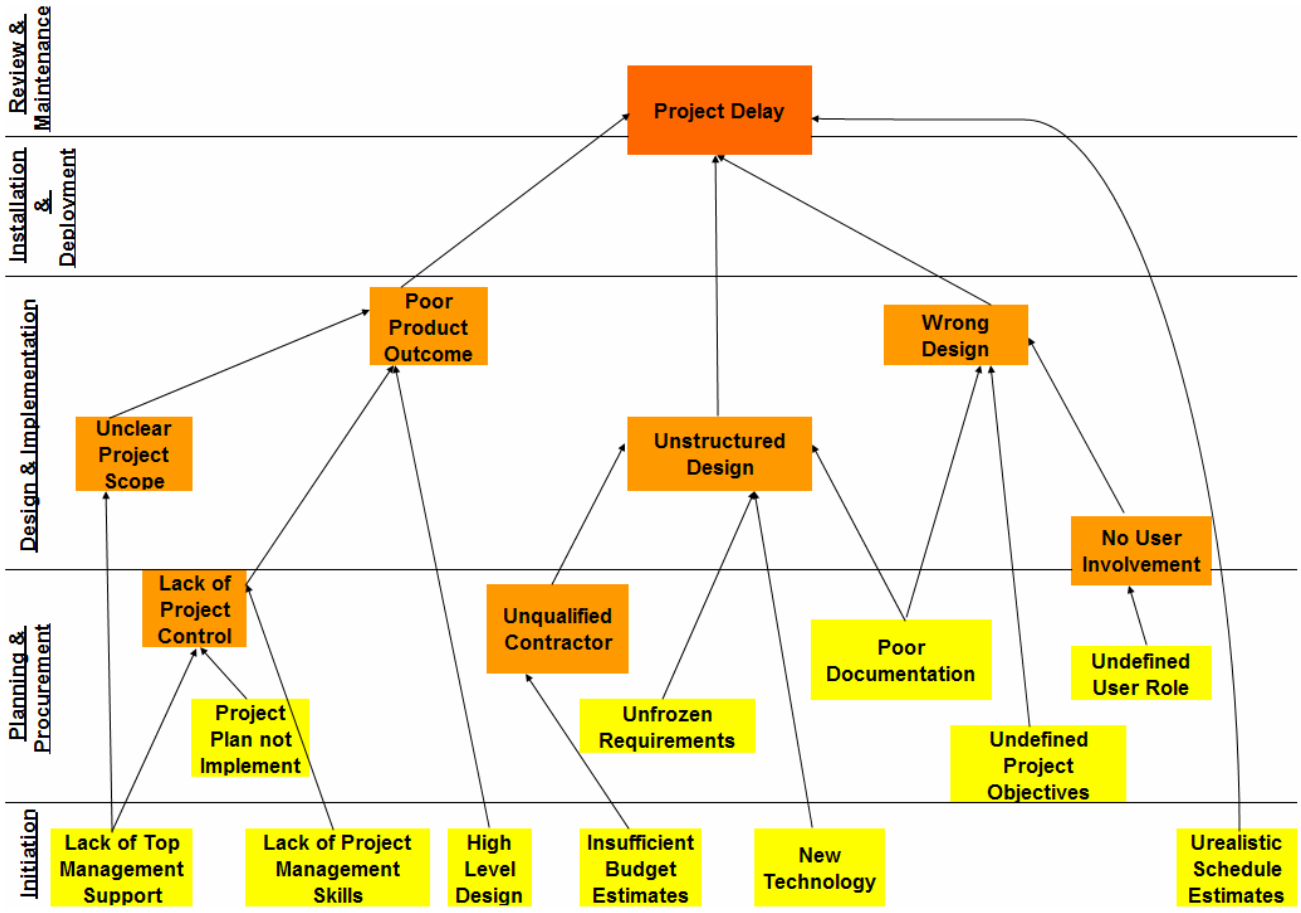

Figure 17: A Causal and Cognitive Map for the Ensuing Case Study [1]

These risks are defined as such:

- Lack of top management support – Risk that the top management will not provide commitment, sponsorship, an end goals perception, an attention to initiatives, funding, and training.
- Project plan not implemented – Risk that the project will not follow the original laid out plan.
- Lack of project management skills – Risk that the project manager or team do not have the appropriate skills for the project most likely due to lack of experience.

- High level design – Risk that the project is extremely complex, that the design is therefore very difficult and tedious.
- Insufficient budget estimates – Risk that the budgets set for this project are extremely optimistic (very low) and that they do not cover the necessities for the completion of the project.
- Unfrozen Requirements – Risk that the requirements of the project will be constantly changing, that they are not stable, 'frozen'
- New technology – Risk that a new technology will be used to develop the project, and since it is new the staff is unfamiliar with it and therefore will have problems developing the project on time or with good quality.
- Poor Documentation – Risk that the documentation of past experiences, and knowledge that needs to be transmitted is very limited or extremely complex (unintelligible).
- Undefined user role – Risk that the developers do not clearly know what will be the role of the users of this software.
- Undefined project objectives – Risk that the developers do not clearly know what the objectives of the project are.
- Unrealistic schedule estimates – Risk that the business aspect of the developers (management) are too optimistic in setting project deadlines.
- Lack of project control – Risk that the project deviates from the original plan and heads in a direction that is undesired.

- Unclear project scope – Risk that the developer can not see what the whole project entails because either the project is too complex or the top management does not express all the details.
- Unqualified contractor – Risk that the developer must hire a contractor that has no previous experience in developing such projects.
- Unstructured design – Risk that the software will be unorganized and therefore difficult to use or understand.
- No user involvement – Risk that the user (client) will not take part of the design of the software, by giving their preferences and opinions.
- Wrong design – Risk that the design of the project is not what the user (client) wants in the end.
- Poor product outcome – Risk that the software even though it does what it is supposed to do, its quality is not very good. It could probably be described as sloppy.

The consequence of all these risks is:

- Project Delay – Risk that the project will not be completed on time

5.2 DEFINE RULES FOR MODEL

Once identified all the risks that afflict the defined project, it is important to develop the interrelationship between these risks in a natural language format. This format is actually what makes this proposed methodology so attractive, since it allows

project managers or project management teams the freedom of expressing all the causes and effects of risk in a given project while not binding them to define exact values. These project managers or project management teams must have some sort of expertise or find experts in this field to define the interrelationship and memberships of the risks they encounter; or they can use the model developed in this thesis, by adding risks, interrelationships, and memberships (in other words by adding new rules), or by manipulating, or even eliminating present rules.

The rules created and developed for this model come from an extensive research into the literature of the risk in the software development industry. Here is a list of the titles of various sources in the literature:

- Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping [1]
- Software Risk Management: Principles and Practice [4]
- A Taxonomy Based Risk Identification [5]
- Risk Analysis in Project of Software Development [14]
- Assessing and Managing the Risks of IS/IT Investment [21]

All of these resources have different perspectives on the risks that affect the software development industry. Some are extremely general, some are extremely specific, very few talk about an interrelationship between risks (they just consider each risk individually), and some feel a few risks have a higher probability of occurring or a higher impact, while others think differently. Therefore, in order to develop the rules that

are used for the model, I analyzed the data from the literature and incorporated my own reasoning.

Samples of the rules developed for the model are presented in the next section and all of the rules can be seen in Appendix A, Appendix B, and Appendix C.

5.3 MODEL AND ALGORITHM DEVELOPMENT

This thesis has the purpose of providing a new tool and a new technique to the software development industry. The algorithm presented here is composed of three models: Project Delay Probability Risk, Project Delay Impact, and Project Attractiveness. The way that the algorithm works is that there are eleven risks that combine and interact with each other contributing to the Project Delay (these risks and the interaction between them are obtained from Al-Shehab, Hughes, Winstanley (2005) and from many other researchers and experts mentioned in the last section). These risks contain two important factors, that of the probability of occurrence and that of the impact these risks have. Therefore two models were created, one to model the interrelationships of the risks contributing to the probability of a Project Delay, and another one to model the interrelationships of the risks contributing to the impact of a Project Delay. These two models return a single value each, the first model returns a probability of Project Delay, and the second one returns the level of impact the Project Delay will cause. A third model was created as well which determines the interrelationship between the probability of a Project Delay and the impact of that delay, returning an output value for the Projects Attractiveness.

A sample of the rules created for the model of Project Delay Probability is shown in Table 1; for the whole set of rules refer to Appendix A. The risks represented in this table are acronyms of the risk factors shown in Figure 17. These acronyms are:

HLD – High Level Design
IBE – Insufficient Budget Estimates
LPC – Lack of Project Control
LPMS – Lack of Project Management Skills
LTMS – Lack of Top Management Support
NT – New Technology
NUI – No User Involvement
PD – Poor Documentation
PPNI – Project Plan Not Implemented
PPO – Poor Product Outcome
UC – Unqualified Contractor
UD – Unstructured Design
UPO – Undefined Project Objectives
UPS – Unclear Project Scope
UR – Unfrozen Requirements
USE – Unclear Schedule Estimates
UUR – Undefined User Role
WD – Wrong Design

Table 1: Project Delay Probability Rules

Rule #	Project Delay Probabily Rules:													Consequence		
2	IF	LTMS	L	OR	PPNI	L	OR	LPMS	M					THEN	LPC	L
4	IF	LTMS	M	AND	PPNI	M	AND	LPMS	M					THEN	LPC	H
8	IF	LTMS	L	OR	HLD	VL								THEN	UPS	VL
12	IF				HLD	H								THEN	UPS	H
13	IF	LTMS	H	AND	HLD	H								THEN	UPS	VH
14	IF	HLD	VH											THEN	UPS	VH
18	IF	IBE	H											THEN	UC	H
19	IF	IBE	VH											THEN	UC	VH
25	IF	UR	H	AND	NT	H	AND	PD	H					THEN	UD	VH
26	IF	UR	VH	OR	NT	VH								THEN	UD	VH
27	IF	UUR	VL											THEN	NUI	VL
28	IF	UUR	L											THEN	NUI	L
33	IF	PD	M	OR	UPO	L	OR	NUI	L					THEN	WD	L
34	IF				UPO	M	OR	NUI	M					THEN	WD	M
35	IF	PD	M	AND	UPO	M	AND	NUI	M					THEN	WD	H
38	IF	UPO	VH	OR				NUI	VH					THEN	WD	VH
39	IF	LPC	VL	OR	UPS	VL	OR	UC	VL	OR	UD	L		THEN	PPO	VL
44	IF	LPC	H	AND	UPS	H	AND	UC	H	AND	UD	H		THEN	PPO	VH
45	IF	LPC	VH	OR	UPS	VH	OR	UC	VH					THEN	PPO	VH
49	IF	PPO	M	AND	UR	M	AND	WD	M	AND	USE	M		THEN	PDelay	H
50	IF	PPO	H	OR				WD	H	OR	USE	H		THEN	PDelay	H
51	IF	PPO	H	AND	UR	H	AND	WD	H	AND	USE	H		THEN	PDelay	VH
52	IF	PPO	VH	OR			OR	WD	VH	OR	USE	VH		THEN	PDelay	VH

What these rules are saying can be shown with a couple of examples from the rules:

- Rule 8: If the probability of the risk that there will be a Lack of Top Management Support is *Low* **OR** that the probability of the risk that the project will be of High Level Design is *VL* **THEN** there is a *Very Low* probability that the risk of an Unclear Project Scope will occur.

- Rule 13: If the probability of the risk that there will be a Lack of Top Management Support is *High* **AND** that the probability of the risk that the project will be of High Level Design is *High* **THEN** there is a *Very High* probability that the risk of an Unclear Project Scope will occur.

A sample of the rules created for the model of Project Delay Impact is demonstrated in Table 2; for the whole set of rules refer to Appendix B. The risks represented in this table are acronyms of the risk factors shown in Figure 17.

Table 2: Project Delay Impact Rules

Rule #	Project Delay Impact Rules:												Consequence		
6	IF	LTMS	H	AND	PPNI	H	AND	LPMS	H				THEN	LPC	VH
7	IF	LTMS	VH	OR				LPMS	VH				THEN	LPC	VH
12	IF	LTMS	H										THEN	UPS	H
13	IF	LTMS	H	AND	HLD	H							THEN	UPS	VH
14	IF	LTMS	VH										THEN	UPS	VH
18	IF	IBE	H										THEN	UC	H
19	IF	IBE	VH										THEN	UC	VH
21	IF	UR	L	OR	NT	L	OR	PD	M				THEN	UD	L
22	IF	UR	M	OR	NT	M							THEN	UD	M
23	IF	UR	M	AND	NT	M	AND	PD	M				THEN	UD	H
24	IF	UR	H	OR	NT	H							THEN	UD	H
27	IF	UUR	VL										THEN	NUI	VL
31	IF	UUR	VH										THEN	NUI	VH
33	IF	PD	M	OR	UPO	L	OR	NUI	L				THEN	WD	L
34	IF				UPO	M	OR	NUI	M				THEN	WD	M
37	IF	PD	H	AND	UPO	H	AND	NUI	H				THEN	WD	VH
38	IF	UPO	VH	OR				NUI	VH				THEN	WD	VH
39	IF	LPC	VL	OR	UPS	L	OR	UC	VL	OR	UD	VL	THEN	PPO	VL
41	IF	LPC	M	OR				UC	M	OR	UD	M	THEN	PPO	M
46	IF	PPO	VL	OR	UR	L	OR	WD	VL	OR	USE	L	THEN	PDelay	VL
47	IF	PPO	L	OR	UR	M	OR	WD	L	OR	USE	M	THEN	PDelay	L
51	IF	PPO	H	AND	UR	H	AND	WD	H	AND	USE	H	THEN	PDelay	VH
52	IF	PPO	VH	OR			OR	WD	VH				THEN	PDelay	VH

What these rules are saying can be shown with a couple of examples from the rules:

- Rule 37: If the impact of the risk that there will be Poor Documentation is *High* **AND** that the impact of the risk that there will be Undefined Project Objectives is *High* **AND** that the impact of the risk that there will be No User Involvement is *High* **THEN** the risk of developing a Wrong Design is a *Very High*.

- Rule 52: If the impact of the risk that there will be Poor Product Outcome is *Very High* **OR** the impact of the risk of developing a Wrong Design is *Very High* **THEN** there is a *Very High* risk that there will be a Project Delay.

A sample of the rules created for the model of Project Attractiveness is demonstrated in Table 3; for the whole set of rules refer to Appendix C. The terms Project Delay Probability, Project Delay Impact, and Project Attractiveness are given the acronyms of PDelayProb, PDelayImpact, and PA respectively.

Table 3: Project Attractiveness Rules

Project Attractiveness									
Rule #		Project Delay Probability			Project Delay Impact			Project Attractiveness	
1	IF	PDelayProb	VL	AND	PDelayImpact	VH	THEN	PA	H
4	IF	PDelayProb	VL	AND	PDelayImpact	L	THEN	PA	VH
5	IF	PDelayProb	VL	AND	PDelayImpact	VL	THEN	PA	VH
8	IF	PDelayProb	L	AND	PDelayImpact	M	THEN	PA	VH
9	IF	PDelayProb	L	AND	PDelayImpact	L	THEN	PA	VH
10	IF	PDelayProb	L	AND	PDelayImpact	VL	THEN	PA	VH
13	IF	PDelayProb	M	AND	PDelayImpact	M	THEN	PA	M
14	IF	PDelayProb	M	AND	PDelayImpact	L	THEN	PA	H
15	IF	PDelayProb	M	AND	PDelayImpact	VL	THEN	PA	VH
16	IF	PDelayProb	H	AND	PDelayImpact	VH	THEN	PA	VL
17	IF	PDelayProb	H	AND	PDelayImpact	H	THEN	PA	L
18	IF	PDelayProb	H	AND	PDelayImpact	M	THEN	PA	L
21	IF	PDelayProb	VH	AND	PDelayImpact	VH	THEN	PA	VL
23	IF	PDelayProb	VH	AND	PDelayImpact	M	THEN	PA	L
25	IF	PDelayProb	VH	AND	PDelayImpact	VL	THEN	PA	H

What these rules are saying can be shown with a couple examples of the rules:

- Rule 4: If the Project Delay Probability is *Very Low* **AND** the Project Delay Impact is *Low* **THEN** the Project Attractiveness is *Very High*.

- Rule 18: If the Project Delay Probability is *High* **AND** the Project Delay Impact is *Medium* **THEN** the Project Attractiveness is *Low*.

The second part of this algorithm is the definition of what is Very Low, Low, Medium, High, and Very High in terms of Risk Probability, Risk Impact, and Project Attractiveness. These values are shown in Table 4.

Table 4: Values of Risk Probability, Impact, and Project Attractiveness

	Risk Probability (percent)	Risk Impact (1-100)	Project Attractiveness (1-100)
Very Low	0 - 23%	0 - 24	0 – 22
Low	18 - 42%	19 - 40	20 – 43
Médium	39 - 63%	37 - 63	38 – 59
High	60 - 84%	58 - 78	57 – 81
Very High	79 - 100%	76 - 100	78 – 100

These ranges of values are different for each model (Risk Probability, Risk Impact, and Project Attractiveness) because they have different characteristics and are each viewed differently. For example, the ranges of temperature are different from those of pressure and those of humidity, because temperature is a different phenomenon than pressure or humidity.

The overlap between Very Low and Low, between Low and Medium, between Medium and High, and between High and Very High allows flexibility between values because for example, a risk of 82% probability of occurring might not be Very High, it might be considered to be High.

5.4 SCENARIOS

Now that the models have been shown and the algorithm explained, a series of scenarios are introduced into the algorithm. The results of these scenarios are presented in graphical form where the first five graphs of the figures created for the Project Delay Probability Model and the Project Delay Impact Model are the Inference Graphs (results from the Fuzzification Process) and the last one is the Composition Graph (composes the Inference Graphs). The defuzzification technique that was used for these models was the centriod method.

Scenario A (Very Low Values of Risk)

Table 5: Scenario A (Very Low Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	8	3
2	PPNI	Project Plan Not Implemented	4	11
3	LPMS	Lack of Project Management Skills	13	21
4	HLD	High Level Design	20	2
5	IBE	Insufficient Budget Estimates	6	17
6	UR	Unfrozen Requirements	5	22
7	NT	New Technology	12	4
8	PD	Poor Documentation	18	17
9	UUR	Undefined User Role	10	4
10	UPO	Undefined Project Objectives	9	2
11	USE	Unrealistic Schedule Estimates	16	23

Scenario A - Project Delay Probability

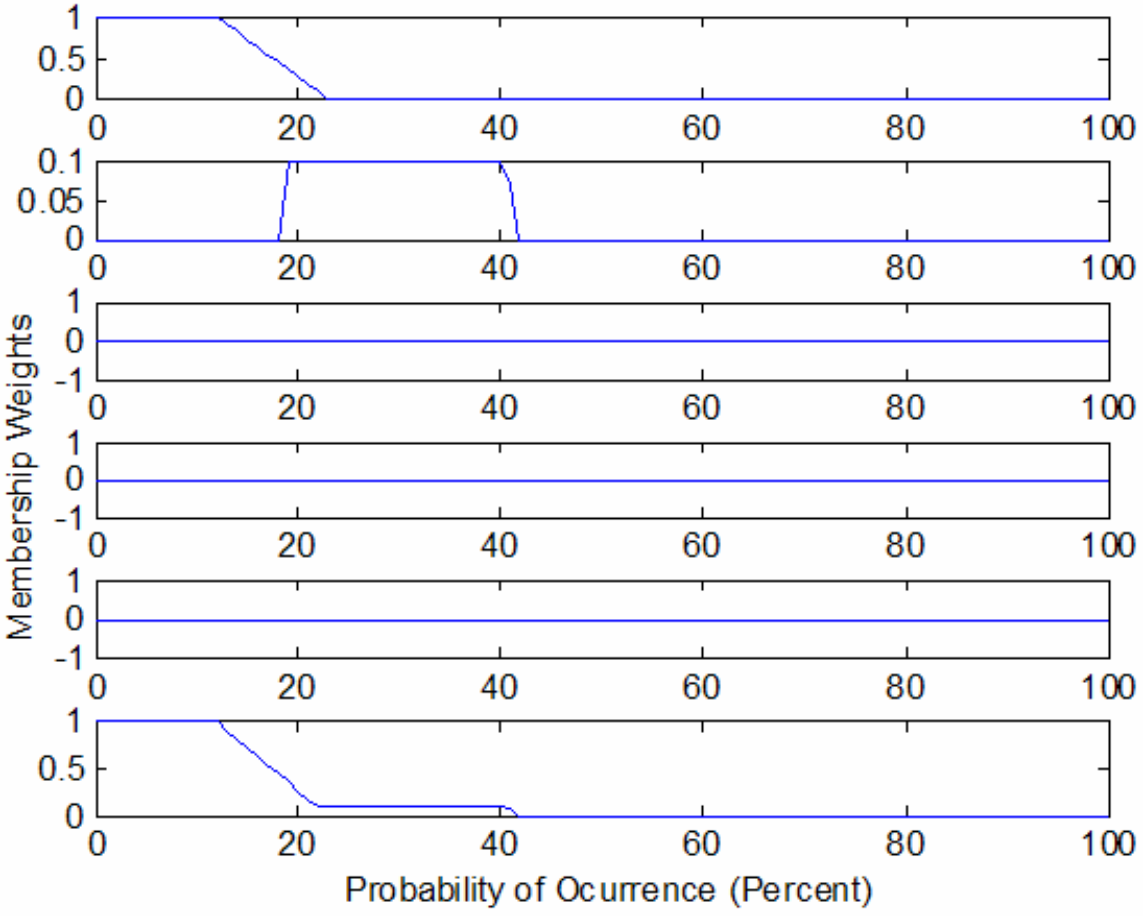

Figure 18: Scenario A - Project Delay Probability

Project Delay Probability – 11%

This value indicates that there is a very low probability of a project delay.

Scenario A - Project Delay Impact

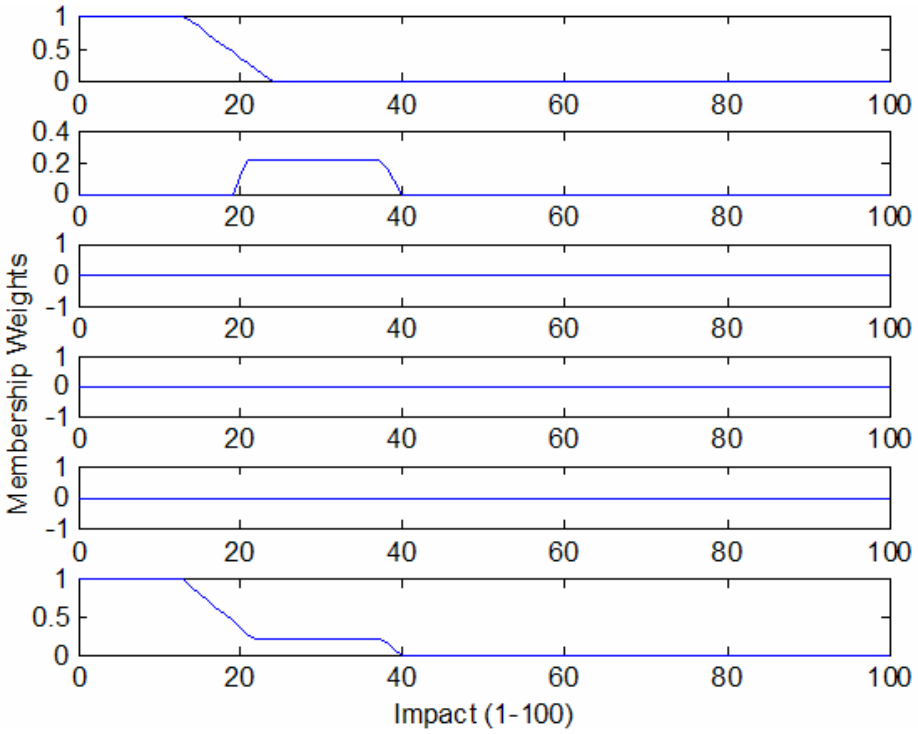

Figure 19: Scenario A - Project Delay Impact

Project Delay Impact – 13

This value indicates that a project delay will have a very low impact on the company.

Scenario A - Project Attractiveness

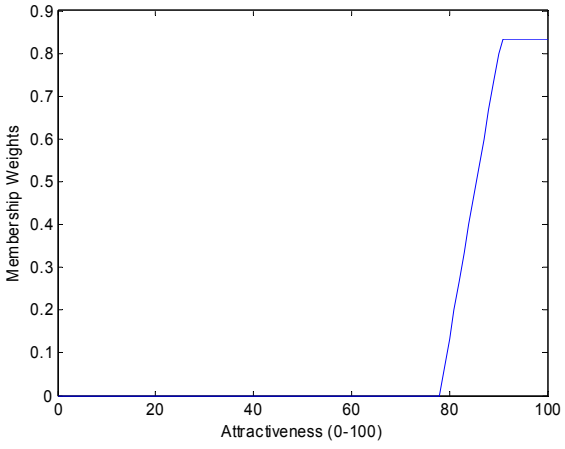

Figure 20: Scenario A - Project Attractiveness

Project Attractiveness – 92

This value indicates the project represented in this scenario is very attractive.

Scenario B (Low Values of Risk)

Table 6: Scenario B (Low Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	33	25
2	PPNI	Project Plan Not Implemented	38	29
3	LPMS	Lack of Project Management Skills	19	28
4	HLD	High Level Design	31	39
5	IBE	Insufficient Budget Estimates	38	27
6	UR	Unfrozen Requirements	30	27
7	NT	New Technology	41	28
8	PD	Poor Documentation	22	26
9	UUR	Undefined User Role	30	30
10	UPO	Undefined Project Objectives	39	25
11	USE	Unrealistic Schedule Estimates	21	29

Project Delay Probability – 17%
Project Delay Impact – 16
Project Attractiveness – 91

Scenario C (Medium Values of Risk)

Table 7: Scenario C (Medium Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	49	58
2	PPNI	Project Plan Not Implemented	55	63
3	LPMS	Lack of Project Management Skills	54	48
4	HLD	High Level Design	58	44
5	IBE	Insufficient Budget Estimates	39	60
6	UR	Unfrozen Requirements	47	62
7	NT	New Technology	45	47
8	PD	Poor Documentation	40	54
9	UUR	Undefined User Role	47	52
10	UPO	Undefined Project Objectives	49	47
11	USE	Unrealistic Schedule Estimates	43	45

Project Delay Probability – 46%
Project Delay Impact – 42
Project Attractiveness – 57

Scenario D (High Values of Risk)

Table 8: Scenario D(High Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	63	62
2	PPNI	Project Plan Not Implemented	68	68
3	LPMS	Lack of Project Management Skills	83	77
4	HLD	High Level Design	64	67
5	IBE	Insufficient Budget Estimates	83	68
6	UR	Unfrozen Requirements	80	66
7	NT	New Technology	75	77
8	PD	Poor Documentation	64	72
9	UUR	Undefined User Role	72	75
10	UPO	Undefined Project Objectives	71	73
11	USE	Unrealistic Schedule Estimates	83	59

Project Delay Probability – 69%
Project Delay Impact – 63
Project Attractiveness – 31

Scenario E (Very High Values of Risk)

Table 9: Scenario E (Very High Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	88	91
2	PPNI	Project Plan Not Implemented	83	91
3	LPMS	Lack of Project Management Skills	84	82
4	HLD	High Level Design	80	89
5	IBE	Insufficient Budget Estimates	87	80
6	UR	Unfrozen Requirements	85	84
7	NT	New Technology	87	88
8	PD	Poor Documentation	94	81
9	UUR	Undefined User Role	83	90
10	UPO	Undefined Project Objectives	88	88
11	USE	Unrealistic Schedule Estimates	85	89

Project Delay Probability – 82%
Project Delay Impact – 91
Project Attractiveness – 10

Scenario F (Combination of Very Low and Low Values of Risk)

Table 10: Scenario F (Combination of Very Low and Low Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	32	7
2	PPNI	Project Plan Not Implemented	3	24
3	LPMS	Lack of Project Management Skills	20	7
4	HLD	High Level Design	22	3
5	IBE	Insufficient Budget Estimates	39	30
6	UR	Unfrozen Requirements	5	16
7	NT	New Technology	29	2
8	PD	Poor Documentation	10	8
9	UUR	Undefined User Role	2	16
10	UPO	Undefined Project Objectives	6	27
11	USE	Unrealistic Schedule Estimates	33	32

Project Delay Probability – 17%
Project Delay Impact – 16
Project Attractiveness – 91

Scenario G (Combination of Very Low, Low, and Medium Values of Risk)

Table 11: Scenario G (Combination of Very Low, Low, and Medium Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	29	32
2	PPNI	Project Plan Not Implemented	10	25
3	LPMS	Lack of Project Management Skills	18	3
4	HLD	High Level Design	45	23
5	IBE	Insufficient Budget Estimates	29	7
6	UR	Unfrozen Requirements	35	42
7	NT	New Technology	58	29
8	PD	Poor Documentation	43	18
9	UUR	Undefined User Role	55	43
10	UPO	Undefined Project Objectives	5	34
11	USE	Unrealistic Schedule Estimates	18	51

Project Delay Probability – 27%
Project Delay Impact – 28
Project Attractiveness – 91

Scenario H (Combination of Very Low, Low, Medium, and High Values of Risk)

Table 12: Scenario H (Combination of Very Low, Low, Medium, and High Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	71	61
2	PPNI	Project Plan Not Implemented	40	57
3	LPMS	Lack of Project Management Skills	60	15
4	HLD	High Level Design	69	73
5	IBE	Insufficient Budget Estimates	69	76
6	UR	Unfrozen Requirements	31	41
7	NT	New Technology	57	54
8	PD	Poor Documentation	56	56
9	UUR	Undefined User Role	2	51
10	UPO	Undefined Project Objectives	83	30
11	USE	Unrealistic Schedule Estimates	57	43

Project Delay Probability – 45%
Project Delay Impact – 37
Project Attractiveness – 69

Scenario I (Combination of Very Low, Low, Medium, High, and Very High Values of Risk)

Table 13: Scenario I (Combination of Very Low, Low, Medium, High, and Very High Values of Risk)

	Acronym	Defined Risks	Probability Values	Impact Values
1	LTMS	Lack of Top Management Support	17	81
2	PPNI	Project Plan Not Implemented	51	67
3	LPMS	Lack of Project Management Skills	6	44
4	HLD	High Level Design	25	92
5	IBE	Insufficient Budget Estimates	64	93
6	UR	Unfrozen Requirements	5	79
7	NT	New Technology	84	34
8	PD	Poor Documentation	41	19
9	UUR	Undefined User Role	13	36
10	UPO	Undefined Project Objectives	32	85
11	USE	Unrealistic Schedule Estimates	83	25

Project Delay Probability – 35%
Project Delay Impact – 50
Project Attractiveness – 92

The first six scenarios are easy to determine without using the Project Attractiveness model, and since the results of the first six scenarios are compatible with our reasoning it is safe to assume that the model of project attractiveness is working correctly. Also it can be observed that when all risks are very high in probability then there is a very high probability of a project delay, and that when all the risks have a high impact then a project delay has a very high impact.

These results show it is easy to choose between a project that is of medium project attractiveness and another that is of high project attractiveness, but it becomes a bit more difficult to choose between projects that both are extremely attractive. This can never be eliminated but it can be fine tuned by adding on more labels to the analysis.

In terms of reasoning this algorithm is working correctly, but in order to fully test this algorithm an extremely complex undertaking has to take place where the algorithm has to test various projects (in the field), past and present, failed and successful. The way to test the algorithm would be to compare the simulated results of Project Attractiveness with various projects that have their risk data available and see if the project results were very bad, bad, average, good, or very good. This analysis though is out of the scope of this project.

Since this algorithm is a new technique in the analysis of risk, it can not be compared with other methodologies in an empirical way, because the interrelationships and weights of the risks have not yet been defined in a standard way.

6 CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The model created in this thesis work is a new method of risk analysis for the purpose of project evaluation by determining the project attractiveness via a qualitative approach. This work provides a contribution to the area of project evaluation since it gives the project manager a simple yet powerful tool that allows him/her to quickly discard projects that are not attractive before doing a complex cash flow analysis or even an ROI (Return On Investment) analysis.

The algorithm created in this thesis work is based upon fuzzy logic, giving this it the ability to solve complex problems plagued with uncertainty and vagueness. Since the software development industry is developing at extremely fast rates, there are lots of risks involved that can affect the outcome of a project and this industry is still not completely adept at dealing with risk. These risks are relatively intangible in nature, since exact values can not be given. This uncertainty makes stakeholders nervous about investing in a new project, which makes it imperative to analyze these risks, but not in the traditional way where specific values are given to the probability of risks to occur and their impact, but in a new way where the stakeholder has a margin of error that will not affect the analysis.

This algorithm allows the user to input values he/she thinks the risk probability and impact is and returns a value for the project's attractiveness. This simplicity makes

this work attractive in the risk analysis area since it permits a fast and effective way of evaluating a project's attractiveness. Also, this algorithm once refined to each area under the industry of software development can be used for subsequent projects, saving large percentages of time, money, and effort, without sacrificing quality.

6.2 RECOMMENDATIONS AND FUTURE WORK

This thesis opens up a realm of possibilities where future researchers can produce a more powerful, user friendly software that can analyze all the possible risk factors with all their specific qualities, producing fast and reliable results.

The particulars that this refinement could entail would be to develop sub-models for each risk where the factors that determine the impact of a risk be considered. These factors could be cost, time, effort, reputation, quality, durability, etc. A Hierarchical Breakdown Structure with the interrelationships between these factors could be determined from studying and working with various on site projects, failed projects, and successful projects. Under the tutelage of experts in the field, the following step would be to refine this algorithm by adapting it to these past and present projects in various areas of the software industry, or add on to this algorithm creating a more robust algorithm that can solve any and all risk analysis problems of the industry.

Another suggestion would be that the cash flows involved in a project be converted into fuzzy logic since generally many aspects of a cash flow are estimates. Once this cash flow is created it would be appropriate to add the impact of the project

attractiveness as a factor in terms of dollars. This feat could most likely be done by assigning a variable constant (a bias) to be multiplied times the value determined (project attractiveness) where the constant will be in relation to the cash flow of a given company (hundreds, thousands, millions).

I would like to suggest another alternative, (most likely a more attractive recommendation than the last one) where the fuzzified cash flow be implemented into the proposed algorithm in this thesis, as a factor of Present Worth or Return On Investment. This incorporation would allow the algorithm to develop from a qualitative approach to a quantitative analysis. The rule set for this new algorithm would look as follows:

IF Probability of Project Delay is ___ and Impact of Project Delay is ___ and Present Worth is ___ THEN Project Attractiveness is ___.

A final recommendation would be to develop the equation of risk exposure, that is seen quite often across the literature, in a fuzzy logic format and taking into consideration the expected value of loss.

REFERENCES

[1] Al-Shehab, Abdullah J. Hughes, Robert T. and Winstanley, Graham "**Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping**". Brighton, UK. The Electronic Journal of Information Systems Evaluation, Vol. 8, Iss. 1, pgs 1-10, 2005.

[2] Baturone, I.; Barriga, A.; Sanchez-Solano, S.; Jiménez-Fernández, C.J.; López, D.R. "**Microelectronic design of fuzzy logic-based systems**". The CRC Press International Series on Computational Intelligence., Boca Raton, page 336, 2000.

[3] Berkan, Riza and Trubatch Sheldon. "**Fuzzy Systems Design Principles Building Fuzzy IF-THEN Rule Bases**". IEEE PRESS, 1997.

[4] Boehm, Barry W. "**Software Risk Management: Principles and Practices**". *IEEE Software*, vol. 8, no. 1, pgs. 32-41, Jan/Feb, 1991.

[5] Carr, Marvin J.; Konda, Suresh L.; Monarch, Ira; Ulrich, Carol F.; Walker, Clay F. "**Taxonomy-Based Risk Identification**". Pittsburgh., PA: .Software Engineering Institute, Carnegie Mellon University, June 1993.

[6] Gallagher, Brian P.; Case, Pamela J.; Creel, Rita C.; Kushner, Susan; Williams, Ray C. "**A Taxonomy of Operational Risks**". Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.

[7] Gray, Clifford F. and Larson, Erik W. "**Project Management: The Managerial Process**". McGraw-Hill. 2nd ed. 2003. pgs. 207-228.

[8] Guo, S. and Peter, L. "**A reconfigurable analog fuzzy logic controller**". IEEE World congress on computational intelligence. Proceeding of the Third IEEE conference on, Vol. 1:pgs. 124-128, June 1994.

[9] Henderson, Susan and Beaumont, Nicholas. "**Managing Information and Communication Technology Project Risk**". Working Paper Series. June 2003. http://www.buseco.monash.edu.au/mgt/research/working-papers/2003/wp43-03.pdf, Downloaded May 2006.

[10] Joaquín, José. "**Design of a Programmable and Modular Analog Fuzzy Controller**". M.S. Thesis. University of Puerto Rico, Mayaguez. 2004.

[11] Kerzner, Harold. "**Project Management: A Systems Approach to Planning, Scheduling, and Controlling**". John Wiley & Sons, Inc. 9th ed. 2006.

[12] Korner, S. Laws of Thought. "**Encyclopedia of Philosophy**". Vol. 4:pgs.414-417, 1967.

[13] Lejewski, C. and Lukasiewicz Jan. "**Encyclopedia of Philosophy"**. Vol. 5: pgs.104-107, NY 1967.

[14] Lu, Xiagnan and Ge, Yalí. "**Risk Analysis in Project of Software Development**". Hangzhou: Zhejiang University Press. IEEE. 2003.

[15] Mahant, Narendra. "**Risk Assessment is Fuzzy Business – Fuzzy Logic Provides the Way to Assess Off-site Risk from Industrial Installations**". Risk 2004. 2004. No. 206.

[16] Mendel, J.M.. "**Fuzzy logic systems for engineering**": A tutorial. Proceedings of the IEEE, Vol. 83(3):pgs. 345-377, March 1995.

[17] Pragya. Agarwal. "**Lotfi Zadeh: Fuzzy logic-incorporating real world vagueness**". Center of Spatially Integrated Social Science. Santa Barbara, CA. 2001-2006.

[18] Ross, Timothy J. "**Fuzzy Logic With Engineering Applications**". McGraw-Hill, New York, New York. 1995.

[19] Sullivan, William G.; Wicks, Elin M.; Luxhoj, James T. "**Engineering Economy**". Pearson Prentice Hall. 13th ed. 2006. pgs. 510-539.

[20] Tah, J.H.M, and Carr, V. A proposal for construction project risk assessment using fuzzy logic. **Construction Management and Economics**. 2000. No. 18. pgs. 491-500.

[21] Ward, John. "**Assessing and Managing the Risks of IS/IT Investments**". Cranfield Working Paper, Bedford, UK. 1992.

[22] Zadeh, L. A.. Fuzzy set. **Information and Control**, Vol. 8(3):338-353, June 1965.

APPENDIX A – Project Delay Probability Rules

Rule #	Project Delay Probabily Rules:													Consequence		
1	IF	LTMS	VL	OR	PPNI	VL	OR	LPMS	L					THEN	LPC	VL
2	IF	LTMS	L	OR	PPNI	L	OR	LPMS	M					THEN	LPC	L
3	IF	LTMS	M	OR	PPNI	M								THEN	LPC	M
4	IF	LTMS	M	AND	PPNI	M	AND	LPMS	M					THEN	LPC	H
5	IF	LTMS	H	OR	PPNI	H								THEN	LPC	H
6	IF	LTMS	H	AND	PPNI	H	AND	LPMS	H					THEN	LPC	VH
7	IF	LTMS	VH	OR	PPNI	VH								THEN	LPC	VH
8	IF	LTMS	L	OR	HLD	VL								THEN	UPS	VL
9	IF	LTMS	M	OR	HLD	L								THEN	UPS	L
10	IF				HLD	M								THEN	UPS	M
11	IF	LTMS	M	AND	HLD	M								THEN	UPS	H
12	IF				HLD	H								THEN	UPS	H
13	IF	LTMS	H	AND	HLD	H								THEN	UPS	VH
14	IF	HLD	VH											THEN	UPS	VH
15	IF	IBE	VL											THEN	UC	VL
16	IF	IBE	L											THEN	UC	L
17	IF	IBE	M											THEN	UC	M
18	IF	IBE	H											THEN	UC	H
19	IF	IBE	VH											THEN	UC	VH
20	IF	UR	VL	OR	NT	VL	OR	PD	L					THEN	UD	VL
21	IF	UR	L	OR	NT	L	OR	PD	M					THEN	UD	L
22	IF	UR	M	OR	NT	M								THEN	UD	M
23	IF	UR	M	AND	NT	M	AND	PD	M					THEN	UD	H
24	IF	UR	H	OR	NT	H								THEN	UD	H
25	IF	UR	H	AND	NT	H	AND	PD	H					THEN	UD	VH
26	IF	UR	VH	OR	NT	VH								THEN	UD	VH
27	IF	UUR	VL											THEN	NUI	VL
28	IF	UUR	L											THEN	NUI	L
29	IF	UUR	M											THEN	NUI	M
30	IF	UUR	H											THEN	NUI	H
31	IF	UUR	VH											THEN	NUI	VH
32	IF	PD	L	OR	UPO	VL	OR	NUI	VL					THEN	WD	VL
33	IF	PD	M	OR	UPO	L	OR	NUI	L					THEN	WD	L
34	IF				UPO	M	OR	NUI	M					THEN	WD	M
35	IF	PD	M	AND	UPO	M	AND	NUI	M					THEN	WD	H
36	IF				UPO	H	OR	NUI	H					THEN	WD	H
37	IF	PD	H	AND	UPO	H	AND	NUI	H					THEN	WD	VH
38	IF	UPO	VH	OR				NUI	VH					THEN	WD	VH
39	IF	LPC	VL	OR	UPS	VL	OR	UC	VL	OR	UD	L		THEN	PPO	VL
40	IF	LPC	L	OR	UPS	L	OR	UC	L	OR	UD	M		THEN	PPO	L
41	IF	LPC	M	OR	UPS	M	OR	UC	M					THEN	PPO	M
42	IF	LPC	M	AND	UPS	M	AND	UC	M	AND	UD	M		THEN	PPO	H
43	IF	LPC	H	OR	UPS	H	OR	UC	H					THEN	PPO	H
44	IF	LPC	H	AND	UPS	H	AND	UC	H	AND	UD	H		THEN	PPO	VH
45	IF	LPC	VH	OR	UPS	VH	OR	UC	VH					THEN	PPO	VH
46	IF	PPO	VL	OR	UR	L	OR	WD	VL	OR	USE	VL		THEN	PDelay	VL
47	IF	PPO	L	OR	UR	M	OR	WD	L	OR	USE	L		THEN	PDelay	L
48	IF	PPO	M	OR				WD	M	OR	USE	M		THEN	PDelay	M
49	IF	PPO	M	AND	UR	M	AND	WD	M	AND	USE	M		THEN	PDelay	H
50	IF	PPO	H	OR				WD	H	OR	USE	H		THEN	PDelay	H
51	IF	PPO	H	AND	UR	H	AND	WD	H	AND	USE	H		THEN	PDelay	VH
52	IF	PPO	VH	OR			OR	WD	VH	OR	USE	VH		THEN	PDelay	VH

APPENDIX B – Project Delay Impact Rules

Rule #	Project Delay Impact Rules:												Consequence		
1	IF	LTMS	VL	OR	PPNI	L	OR	LPMS	VL				THEN	LPC	VL
2	IF	LTMS	L	OR	PPNI	M	OR	LPMS	L				THEN	LPC	L
3	IF	LTMS	M				OR	LPMS	M				THEN	LPC	M
4	IF	LTMS	M	AND	PPNI	M	AND	LPMS	M				THEN	LPC	H
5	IF	LTMS	H				OR	LPMS	H				THEN	LPC	H
6	IF	LTMS	H	AND	PPNI	H	AND	LPMS	H				THEN	LPC	VH
7	IF	LTMS	VH	OR				LPMS	VH				THEN	LPC	VH
8	IF	LTMS	VL	OR	HLD	L							THEN	UPS	VL
9	IF	LTMS	L	OR	HLD	M							THEN	UPS	L
10	IF	LTMS	M										THEN	UPS	M
11	IF	LTMS	M	AND	HLD	M							THEN	UPS	H
12	IF	LTMS	H										THEN	UPS	H
13	IF	LTMS	H	AND	HLD	H							THEN	UPS	VH
14	IF	LTMS	VH										THEN	UPS	VH
15	IF	IBE	VL										THEN	UC	VL
16	IF	IBE	L										THEN	UC	L
17	IF	IBE	M										THEN	UC	M
18	IF	IBE	H										THEN	UC	H
19	IF	IBE	VH										THEN	UC	VH
20	IF	UR	VL	OR	NT	VL	OR	PD	L				THEN	UD	VL
21	IF	UR	L	OR	NT	L	OR	PD	M				THEN	UD	L
22	IF	UR	M	OR	NT	M							THEN	UD	M
23	IF	UR	M	AND	NT	M	AND	PD	M				THEN	UD	H
24	IF	UR	H	OR	NT	H							THEN	UD	H
25	IF	UR	H	AND	NT	H	AND	PD	H				THEN	UD	VH
26	IF	UR	VH	OR	NT	VH							THEN	UD	VH
27	IF	UUR	VL										THEN	NUI	VL
28	IF	UUR	L										THEN	NUI	L
29	IF	UUR	M										THEN	NUI	M
30	IF	UUR	H										THEN	NUI	H
31	IF	UUR	VH										THEN	NUI	VH
32	IF	PD	L	OR	UPO	VL	OR	NUI	VL				THEN	WD	VL
33	IF	PD	M	OR	UPO	L	OR	NUI	L				THEN	WD	L
34	IF				UPO	M	OR	NUI	M				THEN	WD	M
35	IF	PD	M	AND	UPO	M	AND	NUI	M				THEN	WD	H
36	IF				UPO	H	OR	NUI	H				THEN	WD	H
37	IF	PD	H	AND	UPO	H	AND	NUI	H				THEN	WD	VH
38	IF	UPO	VH	OR				NUI	VH				THEN	WD	VH
39	IF	LPC	VL	OR	UPS	L	OR	UC	VL	OR	UD	VL	THEN	PPO	VL
40	IF	LPC	L	OR	UPS	M	OR	UC	L	OR	UD	L	THEN	PPO	L
41	IF	LPC	M	OR				UC	M	OR	UD	M	THEN	PPO	M
42	IF	LPC	M	AND	UPS	M	AND	UC	M	AND	UD	M	THEN	PPO	H
43	IF	LPC	H	OR				UC	H	OR	UD	H	THEN	PPO	H
44	IF	LPC	H	AND	UPS	H	AND	UC	H	AND	UD	H	THEN	PPO	VH
45	IF	LPC	VH	OR				UC	VH	OR	UD	VH	THEN	PPO	VH
46	IF	PPO	VL	OR	UR	L	OR	WD	VL	OR	USE	L	THEN	PDelay	VL
47	IF	PPO	L	OR	UR	M	OR	WD	L	OR	USE	M	THEN	PDelay	L
48	IF	PPO	M	OR				WD	M				THEN	PDelay	M
49	IF	PPO	M	AND	UR	M	AND	WD	M	AND	USE	M	THEN	PDelay	H
50	IF	PPO	H	OR				WD	H				THEN	PDelay	H
51	IF	PPO	H	AND	UR	H	AND	WD	H	AND	USE	H	THEN	PDelay	VH
52	IF	PPO	VH	OR			OR	WD	VH				THEN	PDelay	VH

APPENDIX C – Project Delay Attractiveness Rules

	Project Attractiveness Rules								
Rule #		Project Delay Probability			Project Delay Impact		Consequence		
1	IF	PDelayProb	VL	AND	PDelayImpact	VH	THEN	PA	H
2	IF	PDelayProb	VL	AND	PDelayImpact	H	THEN	PA	H
3	IF	PDelayProb	VL	AND	PDelayImpact	M	THEN	PA	VH
4	IF	PDelayProb	VL	AND	PDelayImpact	L	THEN	PA	VH
5	IF	PDelayProb	VL	AND	PDelayImpact	VL	THEN	PA	VH
6	IF	PDelayProb	L	AND	PDelayImpact	VH	THEN	PA	H
7	IF	PDelayProb	L	AND	PDelayImpact	H	THEN	PA	H
8	IF	PDelayProb	L	AND	PDelayImpact	M	THEN	PA	VH
9	IF	PDelayProb	L	AND	PDelayImpact	L	THEN	PA	VH
10	IF	PDelayProb	L	AND	PDelayImpact	VL	THEN	PA	VH
11	IF	PDelayProb	M	AND	PDelayImpact	VH	THEN	PA	M
12	IF	PDelayProb	M	AND	PDelayImpact	H	THEN	PA	M
13	IF	PDelayProb	M	AND	PDelayImpact	M	THEN	PA	M
14	IF	PDelayProb	M	AND	PDelayImpact	L	THEN	PA	H
15	IF	PDelayProb	M	AND	PDelayImpact	VL	THEN	PA	VH
16	IF	PDelayProb	H	AND	PDelayImpact	VH	THEN	PA	VL
17	IF	PDelayProb	H	AND	PDelayImpact	H	THEN	PA	L
18	IF	PDelayProb	H	AND	PDelayImpact	M	THEN	PA	L
19	IF	PDelayProb	H	AND	PDelayImpact	L	THEN	PA	M
20	IF	PDelayProb	H	AND	PDelayImpact	VL	THEN	PA	H
21	IF	PDelayProb	VH	AND	PDelayImpact	VH	THEN	PA	VL
22	IF	PDelayProb	VH	AND	PDelayImpact	H	THEN	PA	VL
23	IF	PDelayProb	VH	AND	PDelayImpact	M	THEN	PA	L
24	IF	PDelayProb	VH	AND	PDelayImpact	L	THEN	PA	M
25	IF	PDelayProb	VH	AND	PDelayImpact	VL	THEN	PA	H

APPENDIX D – Project Delay Probability Program

```
x = (0:1:100);
VL = Zeta(x,12,23);
L = Triangle(x,18,28,42);
M = Triangle(x,39,53,63);
H = Triangle(x,60,74,84);
VH = Sigmoid(x,79,95);

LTMS = input('Input the value of LTMS: ');
PPNI = input('Input the value of PPNI: ');
LPMS = input('Input the value of LPMS: ');
HLD = input('Input the value of HLD: ');
IBE = input('Input the value of IBE: ');
UR = input('Input the value of UR: ');
NT = input('Input the value of NT: ');
PD = input('Input the value of PD: ');
UUR = input('Input the value of UUR: ');
UPO = input('Input the value of UPO: ');
USE = input('Input the value of USE: ');

LPCVL = max([VL(LTMS) VL(PPNI)
L(LPMS)]);
LPCL = max([L(LTMS) L(PPNI) M(LPMS)]);
LPCM = max([M(LTMS) M(PPNI)]);
LPCH = max([min([M(LTMS) M(PPNI)
M(LPMS)]) H(LTMS) H(PPNI)]);
LPCVH = max([min([H(LTMS) H(PPNI)
H(LPMS)]) VH(LTMS) VH(PPNI)]);

UPSVL = max(L(LTMS),VL(HLD));
UPSL = max(M(LTMS),L(HLD));
UPSM = M(HLD);
UPSH = max([min(M(LTMS),M(HLD))
H(LTMS) H(HLD)]);
UPSVH =
max(min(H(LTMS),H(HLD)),VH(HLD));

UCVL = VL(IBE);
UCL = L(IBE);
UCM = M(IBE);
UCH = H(IBE);
UCVH = VH(IBE);

UDVL = max([VL(UR) VL(NT) L(PD)]);
UDL = max([L(UR) L(NT) M(PD)]);
UDM = max([M(UR) M(NT)]);
UDH = max([min([M(UR) M(NT) M(PD)])
H(UR) H(NT)]);
UDVH = max([min([H(UR) H(NT) H(PD)])
VH(UR) VH(NT)]);

NUIVL = VL(UUR);
NUIL = L(UUR);
NUIM = M(UUR);
NUIH = H(UUR);
NUIVH = VH(UUR);

WDVL = max([L(PD) VL(UPO) NUIVL]);
WDL = max([M(PD) L(UPO) NUIL]);
WDM = max([M(UPO) NUIM]);
WDH = max([min([M(PD) M(UPO) NUIM])
H(UPO) NUIH]);
WDVH = max([min([H(PD) H(UPO) NUIH])
VH(UPO) NUIVH]);

PPOVL = max([LPCVL UPSVL UCVL UDL]);
PPOL = max([LPCL UPSL UCL UDM]);
PPOM = max([LPCM UPSM UCM]);
PPOH = max([min([LPCM UPSM UCM UDM])
LPCH UPSH UCH]);
PPOVH = max([min([LPCH UPSH UCH UDH])
LPCVH UPSVH UCVH]);

PDelayVL = max([PPOVL L(UR) WDVL
VL(USE)]);
PDelayL = max([PPOL M(UR) WDL L(USE)]);
PDelayM = max([PPOM WDM M(USE)]);
PDelayH = max([min([PPOM M(UR) WDM
M(USE)]) PPOH WDH H(USE)]);
PDelayVH = max([min([PPOH H(UR) WDH
H(USE)]) PPOVH WDVH VH(USE)]);

PDelay1 = min(VL,PDelayVL);
PDelay2 = min(L,PDelayL);
PDelay3 = min(M,PDelayM);
PDelay4 = min(H,PDelayH);
PDelay5 = min(VH,PDelayVH);

a=max(PDelay1,PDelay2);
b=max(a,PDelay3);
c=max(b,PDelay4);
finished=max(c,PDelay5);

figure
u=(0:1:100)';
   subplot(611), plot(u, [PDelay1]);
   subplot(612), plot(u, [PDelay2]);
   subplot(613), plot(u, [PDelay3]);
   subplot(614), plot(u, [PDelay4]);
   subplot(615), plot(u, [PDelay5]);
   subplot(616), plot(u, [finished]);

%centroid
Centroid=(sum(x.*finished))/(sum(finished))
```

APPENDIX E – Project Delay Impact Program

```
x = (0:1:100);
VL = Zeta(x,13,24);
L = Triangle(x,19,28,40);
M = Triangle(x,37,52,63);
H = Triangle(x,58,69,78);
VH = Sigmoid(x,76,93);

LTMS = input('Input the value of LTMS: ');
PPNI = input('Input the value of PPNI: ');
LPMS = input('Input the value of LPMS: ');
HLD = input('Input the value of HLD: ');
IBE = input('Input the value of IBE: ');
UR = input('Input the value of UR: ');
NT = input('Input the value of NT: ');
PD = input('Input the value of PD: ');
UUR = input('Input the value of UUR: ');
UPO = input('Input the value of UPO: ');
USE = input('Input the value of USE: ');

%max is OR    %min is AND
LPCVL = max([VL(LTMS) L(PPNI)
VL(LPMS)]);
LPCL = max([L(LTMS) M(PPNI) L(LPMS)]);
LPCM = max([M(LTMS) M(LPMS)]);
LPCH = max([min([M(LTMS) M(PPNI)
M(LPMS)]) H(LTMS) H(LPMS)]);
LPCVH = max([min([H(LTMS) H(PPNI)
H(LPMS)]) VH(LTMS) VH(LPMS)]);

UPSVL = max(VL(LTMS),L(HLD));
UPSL = max(L(LTMS),M(HLD));
UPSM = M(LTMS);
UPSH = max([min(M(LTMS),M(HLD))
H(LTMS)]);
UPSVH =
max(min(H(LTMS),H(HLD)),VH(LTMS));

UCVL = VL(IBE);
UCL = L(IBE);
UCM = M(IBE);
UCH = H(IBE);
UCVH = VH(IBE);

UDVL = max([VL(UR) VL(NT) L(PD)]);
UDL = max([L(UR) L(NT) M(PD)]);
UDM = max([M(UR) M(NT)]);
UDH = max([min([M(UR) M(NT) M(PD)])
H(UR) H(NT)]);
UDVH = max([min([H(UR) H(NT) H(PD)])
VH(UR) VH(NT)]);

NUIVL = VL(UUR);
NUIL = L(UUR);
NUIM = M(UUR);
NUIH = H(UUR);
NUIVH = VH(UUR);

WDVL = max([L(PD) VL(UPO) NUIVL]);
WDL = max([M(PD) L(UPO) NUIL]);
WDM = max([M(UPO) NUIM]);
WDH = max([min([M(PD) M(UPO) NUIM])
H(UPO) NUIH]);
WDVH = max([min([H(PD) H(UPO) NUIH])
VH(UPO) NUIVH]);

PPOVL = max([LPCVL UPSL UCVL UDVL]);
PPOL = max([LPCL UPSM UCL UDL]);
PPOM = max([LPCM UCM UDM]);
PPOH = max([min([LPCM UPSM UCM UDM])
LPCH UCH UDH]);
PPOVH = max([min([LPCH UPSH UCH UDH])
LPCVH UCVH UDVH]);

PDelayVL = max([PPOVL L(UR) WDVL
L(USE)]);
PDelayL = max([PPOL M(UR) WDL
M(USE)]);
PDelayM = max([PPOM WDM]);
PDelayH = max([min([PPOM M(UR) WDM
M(USE)]) PPOH WDH]);
PDelayVH = max([min([PPOH H(UR) WDH
H(USE)]) PPOVH WDVH]);

PDelay1 = min(VL,PDelayVL);
PDelay2 = min(L,PDelayL);
PDelay3 = min(M,PDelayM);
PDelay4 = min(H,PDelayH);
PDelay5 = min(VH,PDelayVH);

a=max(PDelay1,PDelay2);
b=max(a,PDelay3);
c=max(b,PDelay4);
finished=max(c,PDelay5);

figure
u=(0:1:100)';
   subplot(611), plot(u, [PDelay1]);
   subplot(612), plot(u, [PDelay2]);
   subplot(613), plot(u, [PDelay3]);
   subplot(614), plot(u, [PDelay4]);
   subplot(615), plot(u, [PDelay5]);
   subplot(616), plot(u, [finished]);

%centroid
Centroid=(sum(x.*finished))/(sum(finished))
```

APPENDIX F – Project Attractiveness Program

```
x = (0:1:100);
VL = Zeta(x,10,22);
L = Triangle(x,20,31,43);
M = Triangle(x,38,49,59);
H = Triangle(x,57,68,81);
VH = Sigmoid(x,78,93);

PDelayProb = input('Input the value of
PDelayProb: ');
PDelayProb = round(PDelayProb);
PDelayImpact = input('Input the value of
PDelayImpact: ');
PDelayImpact = round(PDelayImpact);

constraint1 =
min(VL(PDelayProb),VH(PDelayImpact));
constraint2 =
min(VL(PDelayProb),H(PDelayImpact));
constraint3 =
min(VL(PDelayProb),M(PDelayImpact));
constraint4 =
min(VL(PDelayProb),L(PDelayImpact));
constraint5 =
min(VL(PDelayProb),VL(PDelayImpact));
constraint6 =
min(L(PDelayProb),VH(PDelayImpact));
constraint7 =
min(L(PDelayProb),H(PDelayImpact));
constraint8 =
min(L(PDelayProb),M(PDelayImpact));
constraint9 =
min(L(PDelayProb),L(PDelayImpact));
constraint10 =
min(L(PDelayProb),VL(PDelayImpact));
constraint11 =
min(M(PDelayProb),VH(PDelayImpact));
constraint12 =
min(M(PDelayProb),H(PDelayImpact));
constraint13 =
min(M(PDelayProb),M(PDelayImpact));
constraint14 =
min(M(PDelayProb),L(PDelayImpact));
constraint15 =
min(M(PDelayProb),VL(PDelayImpact));
constraint16 =
min(H(PDelayProb),VH(PDelayImpact));
constraint17 =
min(H(PDelayProb),H(PDelayImpact));
constraint18 =
min(H(PDelayProb),M(PDelayImpact));
constraint19 =
min(H(PDelayProb),L(PDelayImpact));
constraint20 =
min(H(PDelayProb),VL(PDelayImpact));
constraint21 =
min(VH(PDelayProb),VH(PDelayImpact));
constraint22 =
min(VH(PDelayProb),H(PDelayImpact));
constraint23 =
min(VH(PDelayProb),M(PDelayImpact));
constraint24 =
min(VH(PDelayProb),L(PDelayImpact));
constraint25 =
min(VH(PDelayProb),VL(PDelayImpact));

PA1 = min(H,constraint1);
PA2 = min(H,constraint2);
PA3 = min(VH,constraint3);
PA4 = min(VH,constraint4);
PA5 = min(VH,constraint5);
PA6 = min(H,constraint6);
PA7 = min(H,constraint7);
PA8 = min(VH,constraint8);
PA9 = min(VH,constraint9);
PA10 = min(VH,constraint10);
PA11 = min(M,constraint11);
PA12 = min(M,constraint12);
PA13 = min(M,constraint13);
PA14 = min(H,constraint14);
PA15 = min(VH,constraint15);
PA16 = min(VL,constraint16);
PA17 = min(L,constraint17);
PA18 = min(L,constraint18);
PA19 = min(M,constraint19);
PA20 = min(H,constraint20);
PA21 = min(VL,constraint21);
PA22 = min(VL,constraint22);
PA23 = min(L,constraint23);
PA24 = min(M,constraint24);
PA25 = min(H,constraint25);

a=max(PA1,PA2);
b=max(a,PA3);
c=max(b,PA4);
d=max(c,PA5);
e=max(d,PA6);
f=max(e,PA7);
g=max(f,PA8);
h=max(g,PA9);
i=max(h,PA10);
j=max(i,PA11);
k=max(j,PA12);
l=max(k,PA13);
m=max(l,PA14);
n=max(m,PA15);
```

```
o=max(n,PA16);
p=max(o,PA17);
q=max(p,PA18);
r=max(q,PA19);
s=max(r,PA20);
t=max(s,PA21);
u=max(t,PA22);
v=max(u,PA23);
w=max(v,PA24);
finished=max(w,PA25);

u=(0:1:100)';
   plot(u, [finished]);

%centroid
Centroid=(sum(x.*finished))/(sum(finished))
```

APPENDIX G – Sub Programs

APPENDIX G1 – Sigmoid Sub Program

```
function Sig = Sigmoid(x,min,max)

for i=1:1:length(x)
   if x(i)<=min
      Sig(i)=0;
   end
   if min<=x(i) & x(i)<=max
       Sig(i)=(x(i)-min)/(max-min);
   end
   if x(i)>=max
       Sig(i)=1;
   end
end
```

APPENDIX G2 – Triangle Sub Program

```
function Tri =
Triangle(x,min1,max,min2)

for i=1:1:length(x)
   if x(i)<=min1
      Tri(i)=0;
   end
   if min1<=x(i) & x(i)<=max
      Tri(i)=(x(i)-min1)/(max-min1);
   end
if max<=x(i) & x(i)<=min2
      Tri(i)=(min2-x(i))/(min2-max);
   end
   if min2<=x(i)
      Tri(i)=0;
   end
end
```

APPENDIX G3 – Zeta Sub Program

```
function Z = Zeta(x,max,min)

for i=1:1:length(x)
   if x(i)<=max

      Z(i)=1;
   end
   if max<=x(i) & x(i)<=min

      Z(i)=(min-x(i))/(min-max);
   end
   if x(i)>=min

      Z(i)=0;
   end
end
```