CHAPTER 5

DEVELOPMENT OF A FRAMEWORK USING FUZZY RULE BASE SYSTEMS (FRBS) FOR THE QUANTIFICATION OF COMPLEXITY OF THE SOFTWARE AND ITS USABILITY

The development of FRBS framework starts with Fuzzy Inference System (FIS). The input to FIS may be fuzzy or crisp but the output from FIS is always a fuzzy set as discussed in 3-Block Diagrams of expert systems. The basic step in FIS is to convert the crisp set into fuzzy input. This input is fed to the Rule Base which consists of Knowledge Extractor to generate the complex output set. Later on it is defuzzified to produce the crisp output of a particular event [35, 36]. Since, there is a need to calculate the trade-off value for a software module. The process starts with the basic building block of categorizing the clusters of software module.

5.1 FUZZY RULE BASED SYSTEMS (FRBS)

An Expert System consists of Knowledge Accumulator, Fuzzy Inference System (FIS) and External Variables (Fine Tuning Variables).

Fuzzy Rules Contains linguistic values [46, 47] which are supported by their intensity using IF-THEN-ELSE condition with other linguistic variable. Fuzzy Rule implication can be two kinds of logic inferences: modus ponens and modus

tollens. A simple statement like: "If complexity of software is HIGH, then deliverability is LOW". Since "Unix operating system is complex" according to modus ponens can be infer that "Unix Deliverability is LOW" while according to modus tollens "If Unix operating system is NOT Complex" can be infer that "Unix is Highly Deliverable". FRBS helps in generating a fuzzy model which consists of mapping functionality between set of input variables and set of output variables.

Complex Input problems are simplified in terms of linguistic variables to generate Fuzzy Rule Based. These Rules are matched with GA. These rules iterated and finally refined to eliminate GA generator encodes one complete set of duplicity using Post-Processing stage. Each chromosome generated from Fuzzy Rule.

Each generation was mutated by selecting 20% of the Parent Population, and then these selected individuals were again mutated with a probability of 0.15.

For the above situation, the fitness functions as a component of complexity, deliverability and popularity are obtained.

Finally the Plot as shown in figure 4 which as the popularity of the software (i.e. its deliverability features) increases steadily and plateau at complexity is generated.

$$f = \begin{cases} 1 ; \text{ if Deliverability} < 0 \\ 0.5 ; \text{if Deliverability} > 0 \\ \dfrac{\text{High} - x}{\text{High-Low}} ; \text{if Low} \leq x \leq \text{High ; Complexity} > 0, \text{Popularity} > 0 \\ 2 - \dfrac{\text{Complexity}}{\text{Complexity x Popularity}} ; \quad \text{Popularity} > 0 \\ 0 ; \text{if Complexity} > \text{Deliverability} \end{cases}$$

Equation 9

A general framework of Expert System consists of Knowledge Accumulator Fuzzy Inference System and External Variables (Fine Tuning Variables).

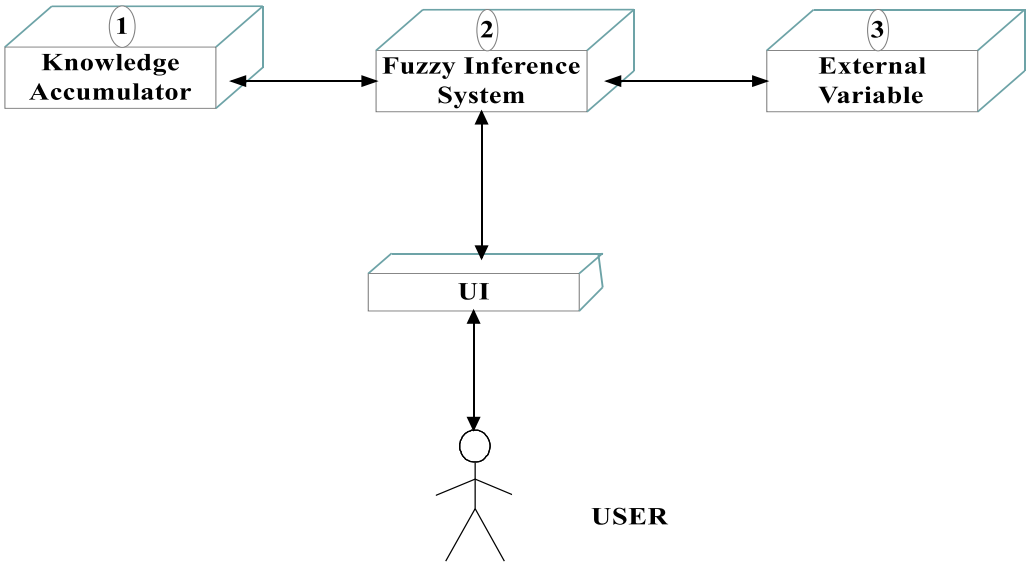

Figure 5.1: Three Block Architecture of Expert System

Knowledge Accumulator gathers knowledge of multiple human experts. This knowledge is fed into the Fuzzy Inference System (FIS) supported by any real time variable. The output is either generated by Sugeno or Mamdani FIS which is converted to the user interface for results and analysis.

5.2 DEVELOPMENT OF THE FRAMEWORK USING FRBS

The development of FRBS framework starts with Fuzzy Inference System (FIS) [51, 52]. The input to FIS may be fuzzy or crisp but the output from FIS is always a fuzzy set as discussed in 3-Block Diagrams of expert systems. The basic step in FIS is to convert the crisp set into fuzzy input. This input is fed to the Rule Base which consists of Knowledge Extractor to generate the complex output set. Later on it is defuzzified to produce the crisp output of a particular event. Since, there is requirement to calculate the trade-off value for a software module. The work

begins with the basic building block of categorizing the clusters of software module.

Earlier identified factors on which FRBS will be applied are:

1. Navigational Guidance
2. Context Shifts
3. System feedback
4. Input parameters
5. New concepts
6. Error feedback

These factors will be the Input for the framework and the usability will be in terms of complexity (High, Medium, and Low)

A simple clustering is achieved by using Fuzzy Decision Tree. From this tree there will be the formation of the Rule base (Figure 3).

Now there are three basic categories of rules:

Rule I: If Complexity is Moderate, Usability is also Moderate, Software is Popular.

Rule II: If Complexity is Very High, Usability is Poor, Software is Not Popular.

Rule III: If Complexity is Low, Usability is High, Software is very Popular.

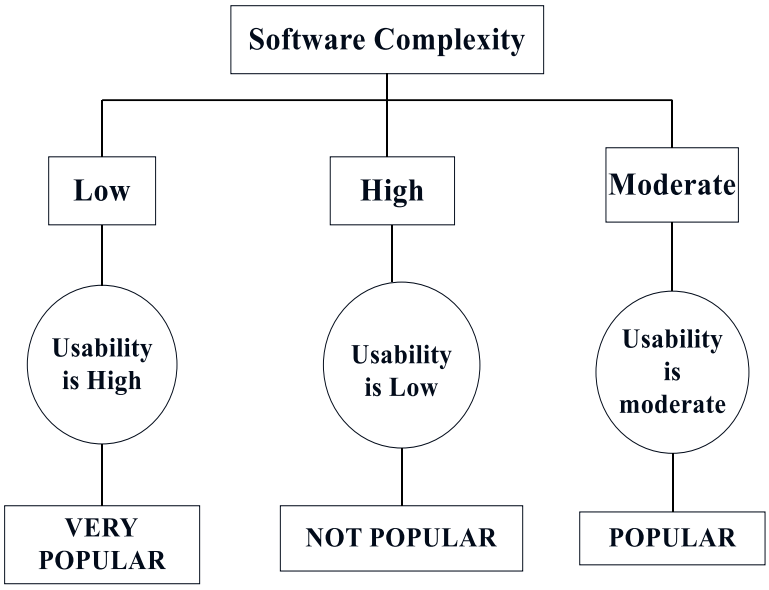

Figure 5.2: Tree hierarchy of Rule-Base

The extension of the basic the above Rule Base of Tree Hierarchy into Fuzzy Inference System in the work is being done.

5.3 QUANTIFICATION OF INPUT VALUES

5.3.1 INPUTS:

1-Complexity (2, 1 and 0 as High, Moderate and Low respectively)

2-Usability (2, 1 and 0 as High, Moderate and Low respectively)

5.3.2 OUTPUT:

Popularity (Ranging from 0 to 10)

The following values has been put into the framework and by applying the three rules that are mentioned above, a framework has been developed which has the input in the form of complexity and usability and the output in the form of popularity.

5.4 FRAMEWORK DEVELOPED USING FRBS

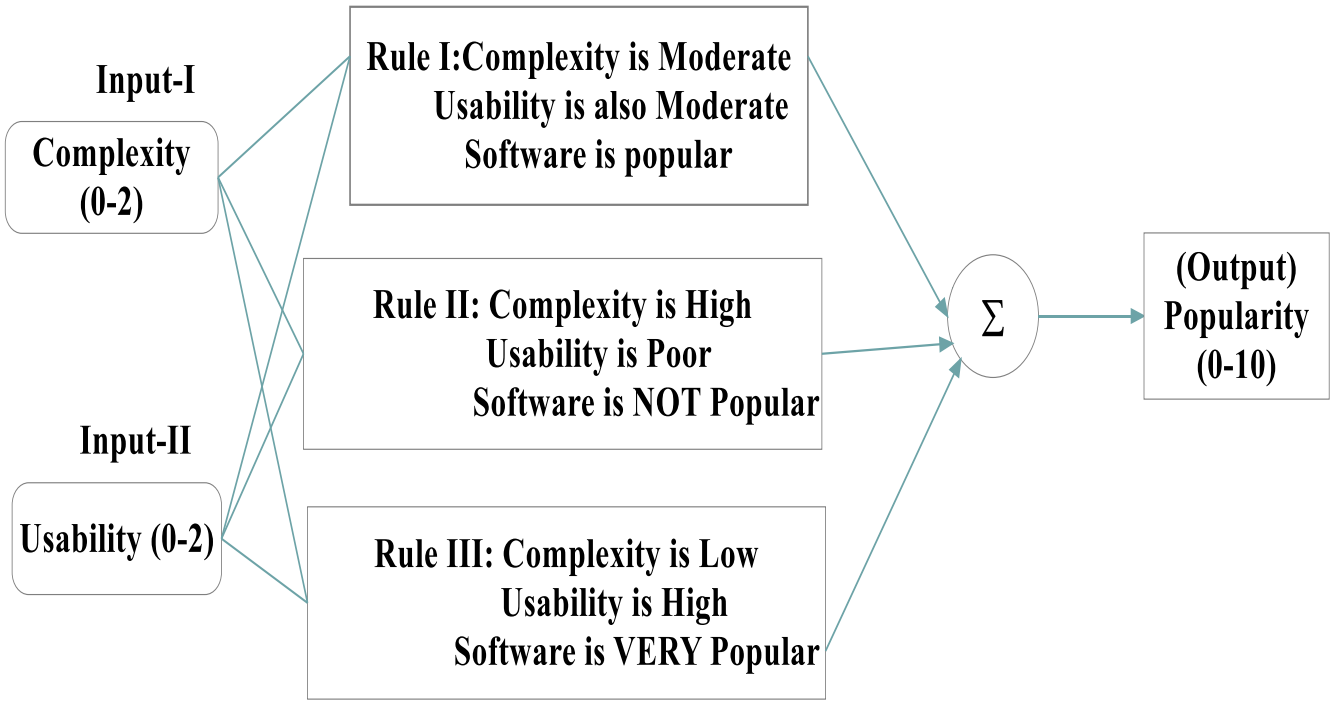

Figure 5.3: Framework developed using FRBS for the quantification of Software complexity and deliverability

The above figure illustrates the rule description of software complexity and usability using Mamdani FIS. Based on the linguistic set, the output is classified into three broad categories: High, Low and Moderate ranging from 0, 1 and 2.

5.5 SURFACE PLOT OF SOFTWARE COMPLEXITY, USABILITY & POPULARITY

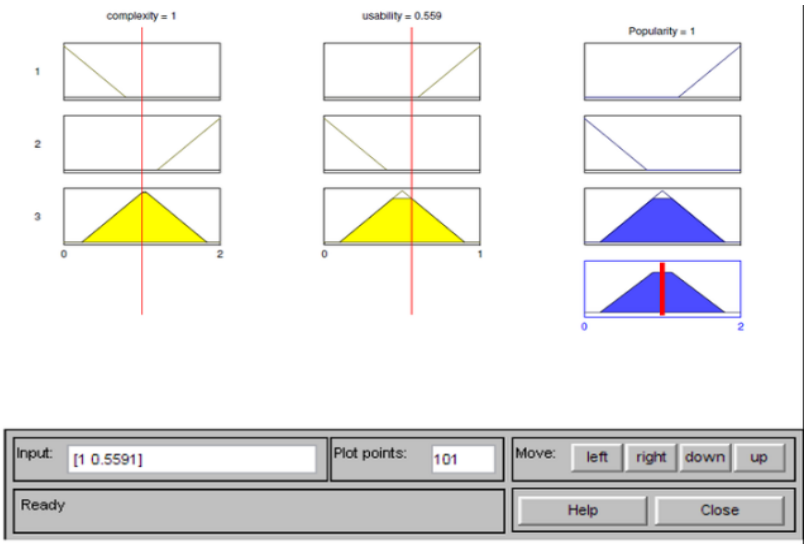

Figure 5.4: Snapshot of deriving a trade-off value of complexity and deliverability

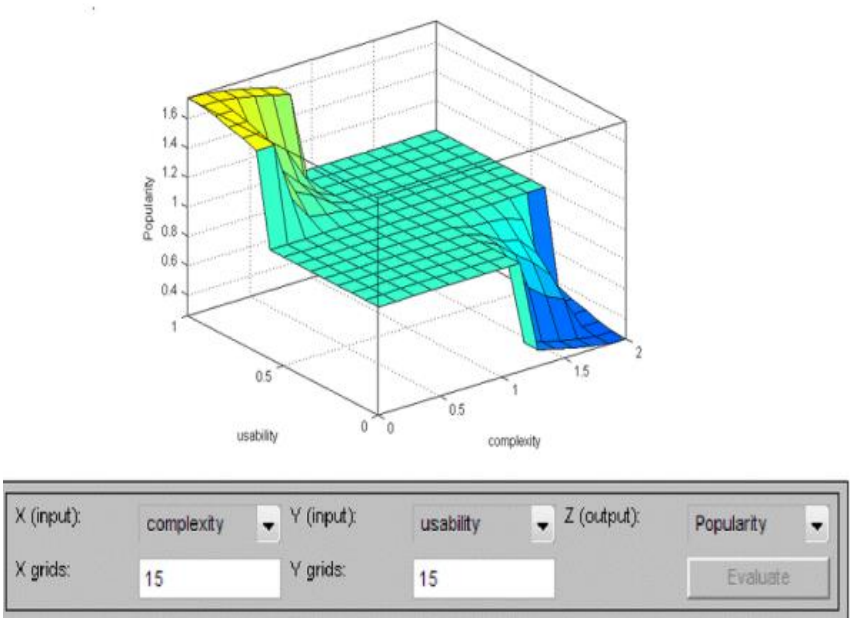

Figure 5.5: Snapshot of Surface plot of software complexity and usability

The surface plot for the above rule base shows the movement of the spike when software complexity is low and usability is high on the VERY popular side of the 3-D graph.

5.6 CONCLUSION

A framework has been developed which takes two inputs (Complexity and Usability) ranging from 0 to 2 and gives the output in the form of popularity ranging from 0 to 10. This framework uses three rules that have been established and quantifies the software attributes like complexity, usability and popularity.