Use of Fuzzy Logic for Software's Risk and Quality Estimation

Ashish Agrawal[1], L.S.Maurya[2]

[1]*M.Tech.(S.E.), S.R.M.S. CET, Bareilly,* [2]*HOD (C.S. Deptt.), S.R.M.S. CET, Bareilly*

Email: agarwal.ashish01@gmail.com, lsmaurya@srmscet.edu

Abstract: Reliability means a system should do what it is supposed to do in an effective and efficient manner. It is one of the major quality attribute of software. Software's reliability get influenced by the factors like requirement specification, risk, time, cost, interaction and resource's capability etc... To measure reliability, while developing or after developing the software; is used to be a tough row to hoe. Though, many software reliability growth models have been evolved so far but still they are restricted to test data and nature of the software project somewhere. In this paper, we propose the use of fuzzy logic with Takagi Sugeno fuzzy system to measure risks and quality for estimating the software's reliability in a better way. as Takagi Sugeno method gives response in fj(x) form that may be linear or constant. We need not to approximate the floating values because fuzzy rules can be applied on floating values.

Keywords: Risk, Reliability, Quality, Fuzzy Models.

I. INTRODUCTION

There is some major quality attributes like usability, portability, correctness, user friendliness to which any software product relates. And reliability is one of them which gives the idea about the percentage or degree up to which the software system meets its expectations and achieve the predefined goals. Reliable software is used to do what it is supposed to do. But it hangs between reliable and unreliable clusters until it is finally deployed in its actual environment. Once software is deployed its performance decides whether it is reliable or unreliable but a prediction about reliability can be made by estimating the risks generated and processes used in developing the particular software product. As society becomes increasingly dependent on systems that in turn depend on the proper functioning of software, experiences with unreliable software become all the more visible and troubling [5] and unreliability may be a result due to severity which as per IEEE (2009) definition is "the highest failure impact that the defect could (or did) cause, as determined by (from the perspective of) the organization responsible for software engineering." This paper is an attempt to make a focus on the use of fuzzy systems in estimating the software's reliability. The results showed in [1] prove that the Takagi-Sugeno fuzzy system was more accurate than the Mamdani system and the LR model for SDEE of projects. And so we are trying and advising to use Takagi Sugeno fuzzy system for risk and quality estimation of software product that leads to reliability prediction of software.

2. LITERATURE SURVEY

In general, there are many risks in the software engineering which is very difficult or impossible to identify all of them. Some of most important risks in software engineering project are categorized as software requirement risks, software cost risks, software scheduling risk, software quality risks, and software business risks. These risks are explained in detail below (Hoodat, H. & Rashidi, H.), table 1 as given shows the various types of risks.. Software reliability is often defined as ―the probability of failure-free operation of a computer program for a specified time in a specified environment. While any system with a high degree of complexity, including software, will be hard to reach a certain level of reliability, system developers tend to push complexity into the software layer, with the rapid growth of system size and ease of doing so by upgrading the software [2].

A. Fuzzy Logic

Fuzzy logic was introduced by L. Zadeh in 1965. The main motivation behind the creation of FL was the existence of imprecision in the measurement process. A fuzzy model is a modeling construct featuring two main properties [3]: (1) It operates at a level of linguistic terms (fuzzy sets that are sets whose elements have degrees of membership), and (2) it represents and processes uncertainty [1].

Fuzzy logic offers a particularly convenient way to generate a keen mapping between input and output spaces thanks to the natural expression of fuzzy rules [4]. The fuzzy inference systems that were used in this comparative study are those proposed by Mamdani and by Takagi-Sugeno. The Mamdani system expects the output membership functions (MF) to be fuzzy sets, whereas the Takagi-Sugeno-type system can be used to model any inference system in which the output is either linear or constant. In this paper we used the constant output [1].

B. Mamdani and Takagi-Sugeno Fuzzy Inference Systems

a) Mamdani system

The rules in standard fuzzy systems (Mamdani) have the form:

R_i : IF x1 is $A^j{}_1$ and …and x_n is $A^j{}_n$ THEN y is B_j

Where R_i denotes the i^{-th} rule, i=1 ,.., Nr , where N_r is the number of rules, x_n is the n^{th} input to the fuzzy system, $A^j{}_n$ and B_j are fuzzy sets described by MF $\mu_{Ai,j}(x_i)$ - [0,1] and $\mu_{Bi}(y)$ - [0,1]. The propositions in the IF part of the rule are combined by applying minimum operators. Sometimes the product is calculated, but it

mostly depends on the situation. The number of prepositions in the consequence part of the rule depends on the number of outputs of the fuzzy system.

b) Takagi-Sugeno system

The rules in functional fuzzy systems (Takagi-Sugeno) have the form:

R_i : IF x1 is $A^j{}_1$ and …and x_n is $A^j{}_n$ THEN y is f_j (x)

Where, $f_j(x)$ is a crisp function of the input variables. For a particular application the effectiveness of the fuzzy system in most cases depends on the order of the function.

3. RISK MANAGEMENT

Risk analysis is often viewed as a “black art”-part fortune telling, part mathematics. Successful risk analysis, however, is nothing more than a business-level decision support tool: it’s a way of gathering the requisite data to make a good judgment call based on knowledge about vulnerabilities, threats, impacts and probability [6].
Risk management is one of the major activities before starting actual developing phase. Effort estimation and risk management become the major issues in the success of software development project, and the accuracy of the results will provide the great support in project execution phase [7]. Risk management identifies addresses, prioritizes, eliminates and mitigates the software risks before they make any undesired and great loss. It helps the developers in making an idea about the success or failure of the software in future.
Software risks affect the reliability, cost, schedule and the quality of the products.

Table 1- Type of Software Risks

SOFTWARE REQUIREMENT RISKS	1. Lack of analysis for change of requirements. 2. Change extension of requirements 3. Lack of report for requirements 4. Poor definition of requirements 5.Ambiguity & change of requirements
SOFTWARE COST RISKS	1.Lack of good estimation in projects 2. Unrealistic schedule 3.The hardware does not work well
SOFTWARE SCHEDULING RISKS	1. Inadequate budget 2. Human errors 3.Inadequate knowledge about tools and techniques 4. Long-term training for

	personnel
SOFTWARE QUALITY RISKS	1.Inadequate documentation 2. Lack of project standard & estimation 3. Lack of design documentation 4. Inadequate budget 5. Human errors 6.Unrealistic schedule

4. IMPLEMENTATION AND RESULTS

In this research, for estimating risk we took reference from the Madachy risk assignment matrix (Madachy, 97). For generating this matrix, we took two attributes *Size* and *Schedule (SCDE)* as inputs and *Risk* as output. The resultant matrix is shown in table 2. With the help of this matrix we developed some fuzzy if-then rules which were applied to both the Mamdani Inference Model and the Takagi Sugeno Model of fuzzy model in Mat lab (R2009b). For fuzzy, the range was [0,1]. Both the models are described in previous sections. Figure 1 shows the algorithm for performing the fuzzy logic on software data.

A. Rule Editor

In fuzzy product of Mat lab, the rule viewer is used to make new or to update the rules of a FIS structure and the file is saved as 'a.fis'. For creating the rules, we must firstly define the input and output variables then we can create the rules with the help of checkboxes, list boxes, connections and weights. Here, for risk estimation we use same rules for both Mamdani and Takagi Sugeno model in rule editor, as shown in fig. 2.

B. Rule Viewer

The Rule Viewer invoked using rule view ('a') depicts the fuzzy inference diagram for

Table 2 Size vs. Schedule Risk Assignment Matrix

	Schedule				
	Very Low	**Low**	**Nominal**	**High**	**Very High**

	Very Low					
	Low	Very Low				
Size	**Nominal**	Low	Very Low			
	High	Moderate	Low	Very Low		
	Very High	High	Moderate	Low	Very Low	
	Extra High	Very High	High	Moderate	Low	Very Low

a FIS stored in a file, a.fis. We can move around the line indices that correspond to the inputs and watch the system readjust and compute the new output [8]. Figure 3 shows the rule viewer of Mamdani model and fig.4 shows the rule viewer of Takagi Sugeno Model, after applying the If-then rules.

C. Surface Viewer

The Surface Viewer, invoked using *surfview('a')*, is a GUI tool that lets you examine the output surface of a FIS stored in a file, a.fis, for any one or two inputs. Because it does not alter the fuzzy system or its associated FIS structure in any way, Surface Viewer is a read-only editor. Using the drop-down menus, you select the two input variables you want assigned to the two input axes (X and Y), as well the output variable you want assigned to the output (or Z) axis. If you want to create a smoother plot, use the Plot points field to specify the number of points on which the membership

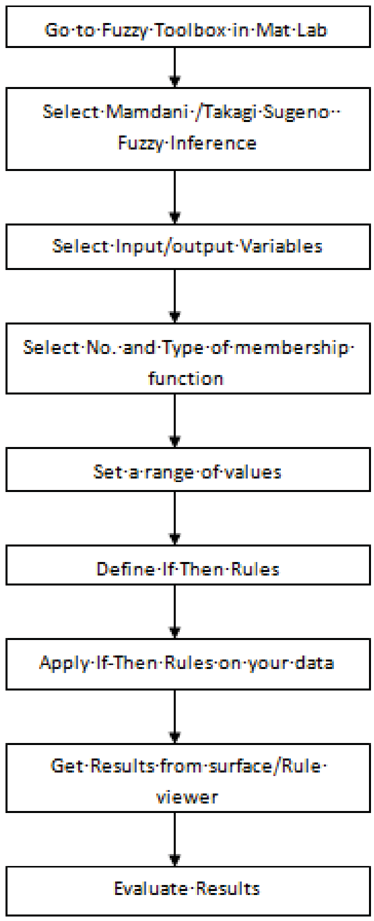

Fig 1 Algorithm for performing fuzzy logic in MatLab

functions are evaluated in the input or output range. This field defaults to a value of 101[8].

Figure 5 shows the surface viewer of Mamdani model and fig.6 shows the surface viewer of Takagi Sugeno Model, after applying the If-then rules

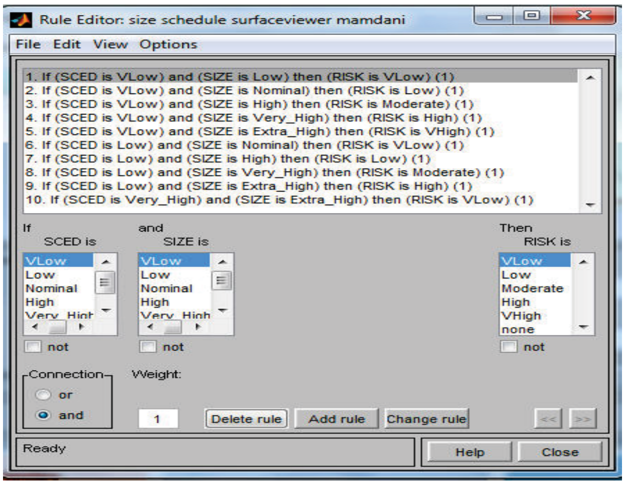

Fig. 2 Rules Editor with If-Then Rules

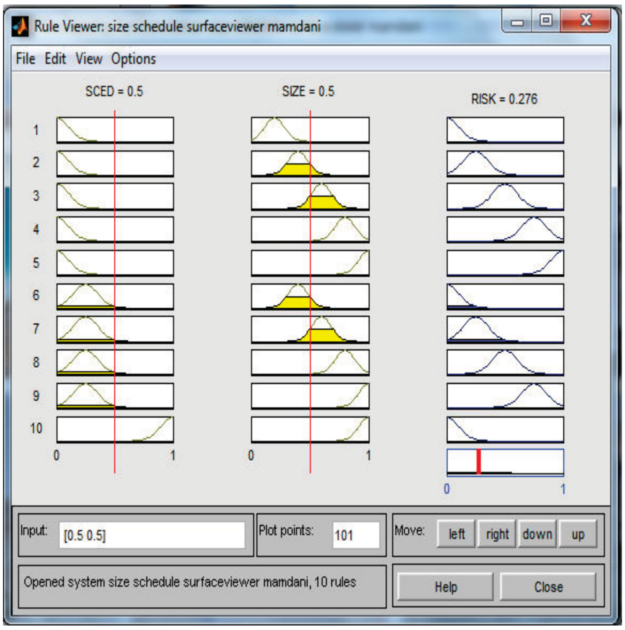

Fig. 3 Rule Viewer for Mamdani system

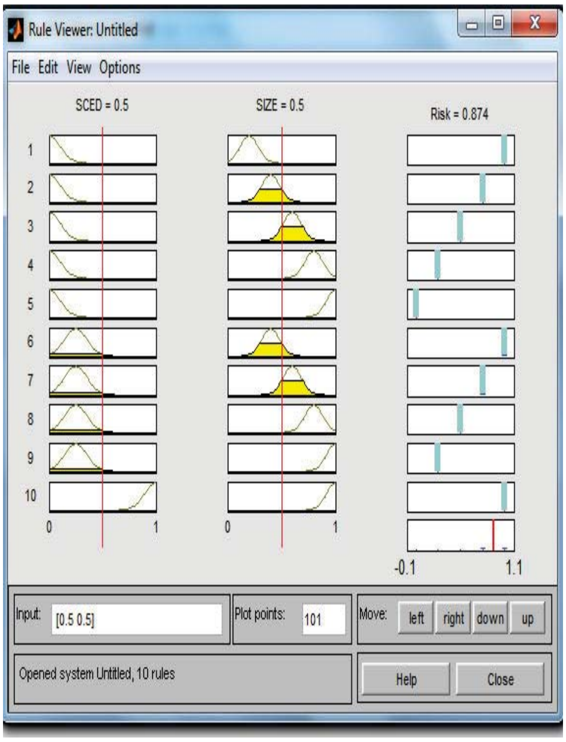

Fig. 4 Rule Viewer for Takagi Sugeno Model

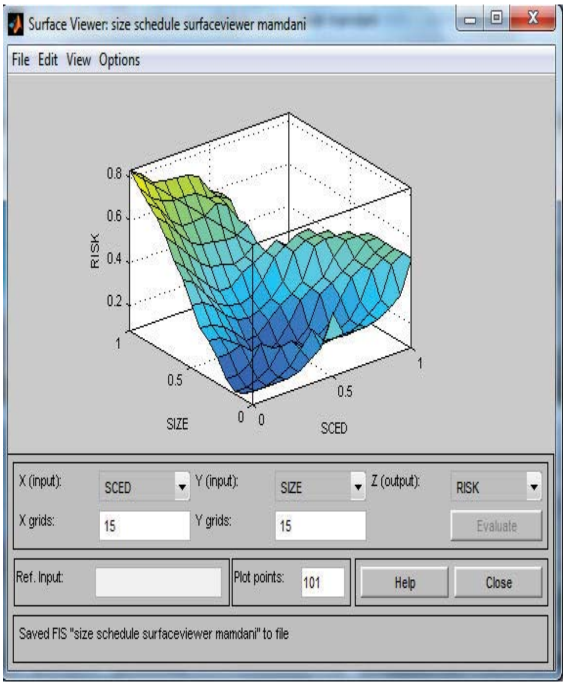

Fig. 5 Surface Viewer for Mamdani Model

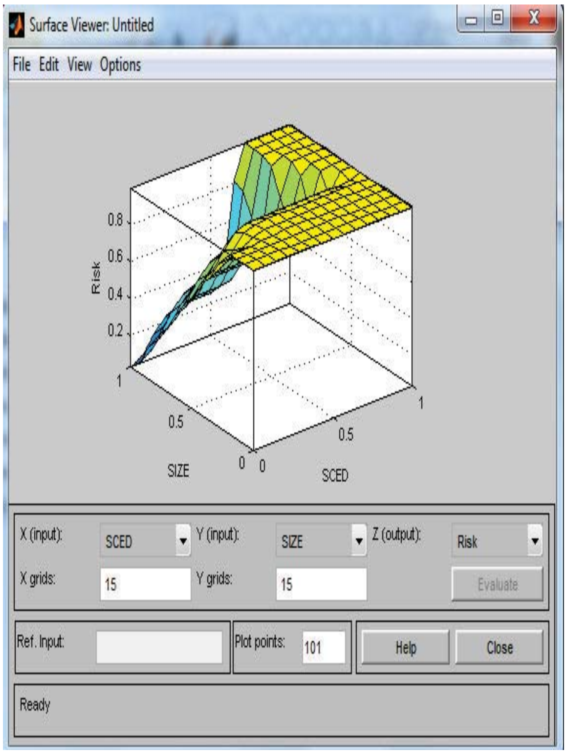

Fig. 6 Surface Viewer for Takagi Sugeno Model

5. CONCLUSION

By using fuzzy logic principles, we once define the if then rule base then we can check it for multiple values instead of defining rules again and again. At the same time, the results can be evaluated in 3D projection by surface viewer and in 2D projection by rule viewer. With this research, we conclude that fuzzy systems can be a way for estimating the risk of software because fuzzy system can deal with crisp values of risk as well as floating values of risk.

Figure 2 shows if the schedule value is 0.5 and size is also 0.5 than on the basis of the rule base, Mamdani fuzzy inference gives the estimated risk value as 0.276 whereas for

Takagi Sugeno inference, this value is 0.874.

In future, Takagi Sugeno model can be applied with linear operation for estimating the risk and reliability and that may result better. The attributes for risk estimation can be cost, changing requirements, lack of resources.

References

[1] Noel Garcia-Diaz, Cuauhtemoc Lopez-Martin, Arturo Chavoya "A comparative study of two fuzzy logic models for software development effort estimation", Published by Elsevier Ltd.,2013

[2] Bonthu Kotaiah , Dr. R.A. Khan , "A survey on software reliability assessment by using different machine learning techniques", International Journal of Scientific & Engineering Research, Volume 3, Issue 6, June-2012 , ISSN 2229-5518.

[3] Zhiwei Xu Z. and Khoshgoftaar T. M., 2004. "Identification of fuzzy models of software cost estimation". Elsevier Fuzzy Sets and Systems. pp.141-163.

[4] Zadeh L.A., 2002. "From computing with numbers to computing with words–from manipulation of measurements to manipulation of perceptions". Journal AMCS. Pp. 307-324.

[5] Taz Daughtrey**, "**The many faces of software unreliability", SQP VOL. 14, NO. 4/© 2012, ASQ,pp 4-12.

[6] Gary McGraw, "Risk analysis in software design", The IEEE Society, 2004.

[7] Pressman, R.S. (2005), "Software Engineering-practitioner's approach" (6^{th} edition).New York: McGraw Hill

[8] http://www.mathworks.in/help/fuzzy