Volume : 1 | Issue : 5 | Nov 2012 • ISSN No 2277 - 8160

Research Paper | Computer science

Study of Fuzzy Logic Approach in Estimating Reliability of Component Based Software

Harish Rathod Dept. of CSE/IT, B. H. Gardi College of engineering & Technology, Rajkot, India

Mahesh Parmar Dept. of CSE/IT, B. H. Gardi College of engineering & Technology, Rajkot, India

ABSTRACT

With the great development of computer technology, software designers are motivated to integrate commercial of the shelf software components for rapid software development. To ensure high reliability for such applications using software components as their building blocks to construct a software system, dependable components have to be deployed to meet the reliability requirements [1]. The last decade has shown that object-oriented concept by itself is not that powerful to cope with the rapidly changing requirements of ongoing applications. Component Based Software achieves flexibility by clearly separating the stable parts of systems (i.e. the components) from the specification of their composition. It becomes necessary to ensure the reliability of components as well the reliability of overall system. This paper presents a new approach to analyze the reliability of component-based software, based on the reliabilities of the individual components. In order to realize the reuse of components effectively in Component Based Software Development, it is required to measure the reliability of components. In this paper, we adopt Fuzzy Logic approach to estimate the reliability of component.

KEYWORDS :

I. Introduction to CBSE

Component Based Software Engineering (CBSE) is a paradigm that aims at constructing and designing systems using a pre-defined set of software components explicitly created for reuse. component-based software development is most promising approach for software development today. This approach is based on the idea that software systems can be developed by selecting appropriate off-the-shelf components and then assembling them with well-defined software architecture [2]. This new software development approach is very different from the traditional approach in which software systems can only be implemented from scratch.

Reliability: Software reliability is defined as the probability of failure –free software operation for a specified period of time in a specified environment. The reliability of a software product is usually defined to be "the probability of execution without failure for some specified interval of natural units or time" [3]. This is an operational measure that varies with how the product is used. Reliability of a component is measured in the context of how the component will be used. That context is described in an operational profile. Reliability of a piece of software may be computed or measured. If the software has not been built yet, its reliability can be computed from a structural model and the reliabilities of the individual parts that will be composed to form the software. There is error associated with this technique due to emergent behaviors, i.e. interaction effects, which arise when the components must work together. If the software is already assembled, the reliability can be measured directly. The measurement is taken by repeatedly executing the software over a range of inputs, as guided by the operational profile. The test results for the range of values are used to compute the probability of successful execution for a specific value. The error associated with this approach arises from the degree to which the actual operation deviates from the hypothesized operation assumed in the operational profile.

II. What is reliability?

"The probability that an item will perform a required function without failure under stated conditions for a stated period of time" [4], Reliability refers to the consistency of a measure. A test is considered reliable if we get the same result repeatedly. For example, if a test is designed to measure a trait (such as introversion), then each time the test is administered to a subject, the results should be approximately the same. Unfortunately, it is impossible to calculate reliability exactly, but there are several different ways to estimate reliability.

According to ANSI, Software Reliability is "the probability of failure-free software operation for a specified period of time in a specified environment". Software Reliability is also an important factor affecting system reliability [5].

III. FUZZY LOGIC

Fuzzy set theory proposed in 1965 by Lofti A. Zadeh is a generalization of classical set theory. Fuzzy set theory tries to capture the way humans represent and reason with real-world [7]. Fuzzy Logic is a multi valued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, high/low, etc. Notions like rather tall or very fast can be formulated mathematically and processed by computers, in order to apply a more human-like way of thinking in the programming of computers. Fuzzy logic is a methodology to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory. Use of fuzzy sets in logical expression is known as FL that has been the subject of important investigations. In the beginning of the nineties, FL was firmly grounded in terms of its theoretical foundations and used in various fields. The central assertion underlying this approach is that entities in the real world simply do not fit into neat categories. For example, a project is not small, medium, or large. It could in fact be something in between; perhaps mostly a large project but also something likes a medium project. This can be represented as a degree of belonging to a particular linguistic category.

Consider the query, "Is water colorless?" The answer to this is a definite Yes/True, or No/False, as warranted by the situation. If "yes"/"true" is accorded a value of 1 and "no"/"false" is accorded a value of 0, this statement results in a 0/1 type of situation. Such a logic which demands a binary (0/1) type of handling is termed crisp in the domain of fuzzy set theory. Thus, statements such as "Temperature is 32 C", "The running time of program is 4 seconds" are examples of crisp situations.

On the other hand, consider the statement, "Is Ram honest?" The answer to this query need not be definite "yes" or "no". Considering the degree to which one knows Ram, a variety of answers spanning a range, such as "extremely honest", "extremely dishonest", "honest at times", "very honest" could be generated. If, for instance, "extremely honest" were to be accorded a value of 1, at the high end of the spectrum of values, "extremely dishonest" a value of 0 at the low end of the spectrum, then, "honest at times" and "very honest" could be assigned values of 0.4 and 0.85 respectively. The situation is therefore so fluid that it can accept values between 0 and 1, in contrast to the earlier one which was either a 0 or 1. Such a situation is termed fuzzy. Figure 1, shows a simple diagram to illustrate fuzzy and crisp situations.

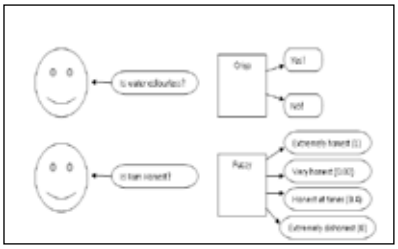

Figure: 1 Fuzzy versus crisp

Fuzzy set theory can be defined mathematically by assigning to each possible individual in the universe of disclosure, a value representing its grade of membership in the fuzzy set. Fuzzy sets can be effectively used to represent linguistic values such as low, young, and complex. Major advantage of this approach is that it is less dependent on historical data. A FL system which accepts imprecise data and vague statements such as low, medium, high and provides decisions as shown in Figure 2, implementing a fuzzy system requires that the different categories of the different inputs be represented by fuzzy sets which, in turn, are represented by membership functions. The domain of membership function is fixed, usually the set of real numbers, and whose range is the span of positive numbers in the closed interval [0, 1].

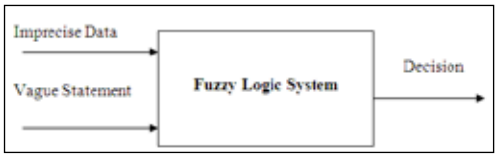

Figure: 2 Fuzzy Logic Systems

Fuzzy rule-based systems apply to solve many types of "real-world" problems, especially where a system is difficult to model or controlled by a human operator or expert, or where ambiguity or vagueness is common. A typical fuzzy system consists of a rule base, Membership functions and an inference procedure.

Fuzzy inference systems have been successfully applied in the fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems.

Mamdani's fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani [8] as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani›s effort was based on Lotfi Zadeh›s 1973 paper on fuzzy algorithms for complex systems and decision processes [9].

IV. Parameters of Fuzzy Logic

The parameters of Fuzzy Logic are as follows:
Fuzzy Sets: Fuzzy sets support a flexible sense of membership of elements to a set. While in crisp set theory, an element either belongs to or does not belongs to a set, in fuzzy theory many degree of membership (between 0 and 1) are allowed.

Thus, a membership function $\mu\tilde{A}(x)$ is associated with a fuzzy set $\tilde{A}$ such that the function maps every element of the universe of discourse x to the interval [0,1].

Types of fuzzy sets are defined as follows:

- Union ($A \wedge B$)
- Intersection ($A \vee B$)

Complement (A)

Linguistic Variable: It represents linguistic values such as low, young, and complex [7]. Membership Function: The only condition a membership function must really satisfy is that it must vary between 0 and 1. The membership function need not always be described by desecrate values A membership function can also be given mathematically as

$\mu\tilde{A}(x)=1/(1+x)*(1+x)$ (1)

Different shapes of membership function exist. The shape could be Triangular, Trapezoidal, and Gaussian.

• **Fuzzy Control:** Fuzzy Control combines the use of fuzzy linguistic variables with fuzzy logic. For example: Speed Control i.e. how fast am I going to drive today?

• **Fuzzification:** Fuzzification is the process of calculating the input membership functions.

• **Defuzzification:** The conversion of a fuzzy set to single crisp value is called Defuzzification and is the reverse process of Fuzzification. Several methods are available [7] for Defuzzification, namely centroid method, center of sums and mean of maxima.

V. Existing techniques

In component based software system, if a system consists of n components with reliabilities denoted by R1,…, Rn respectively, the reliability of system is an execution path, 1, 3, 2, 3, 2, 3, 4, 3, n, is given by Rs. Thus, the objective here is to estimate the reliability of a system by averaging over all path reliabilities. For this propose system consider the architecture of software as shown in Fig 3. Assume that, the application consists of n components, and has a single initial state denoted by 1, and a single absorbing or exit state denoted by n.
The expected reliability of system is defined by the following equation:
The expected reliability of system is defined by the following equation:
E[Rs] = E[] (2)

Where:
E[Rs] is the estimated reliability of the system.
Rim1,i is the reliability of individual component.

Thus to obtain the expected reliability of the application, we need to obtain E[Rim1,i] which is the expected reliability of component i for a single execution. From the Taylor series expression of the function of a random variable [6] we have:
E[Rim1,i]= + (3)

Where: Var[m1,i] is the variance of individual component.
Since the number of visits to the absorbing state n is always 1, E[m1,n] = 1 and Var[m1,n] = 0 and hence E[Rnm1,n] = Rn.
Equation (3) can be written as:
E[Rs]=[Πi=1to n-1 +] Rn (4)

The result is shown in Table 2 according to equation (3) and (4)

p1,2 = 0.60	p1,3 = 0.20	p1,4 = 0.20	
p2,3 = 0.70	p2,5 = 0.30		
p3,5 = 1.00			
p4,5 = 0.40	p4,6 = 0.60		
p5,7 = 0.40	p5,8 = 0.60		
p6,3 = 0.30	p6,7 = 0.30	p6,8 = 0.10	p6,9 = 0.30
p7,2 = 0.50	p7,9 = 0.50		
p8,4 = 0.25	p8,10 = 0.75		
p9,8 = 0.10	p9,10 = 0.90		

Table 1: Transaction probability of components

The system reliability is 0.8267 using equation (3).

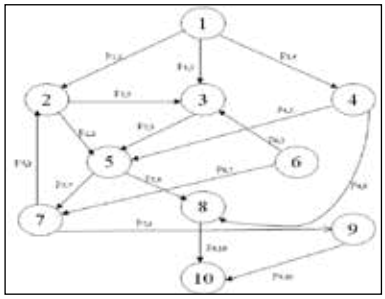

Figure: 3 Architecture of Component Based Software

Component No.	Assume Reliability (R_i)	Mean (m)	Variance (v)	Estimated Reliability (R_i)
1	0.9990	1.0000	0.0000	0.9990
2	0.9800	0.9077	0.6444	0.9819
3	0.9900	0.9107	0.5499	0.9909
4	0.9700	0.4184	0.3928	0.9874
5	0.9500	1.3504	0.7185	0.9337
6	0.9950	0.2510	0.2319	0.9987
7	0.9850	0.6155	0.6261	0.9908
8	0.9500	0.8737	0.4255	0.9564
9	0.9750	0.3831	0.2462	0.9904
10	0.9850	1.0000	0.0000	0.985

Table 2: Estimated reliability using existing technique

VI. Proposed Methodology

Fuzzy logic is a methodology to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory, which introduced by Prof. Zadeh in 1965. Here we propose Fuzzy Logic model to estimate the reliability of component based software system as shown in Fig 4.

We will use following parameters for estimating reliability of Component Based Software using Fuzzy Logic.

- Input fuzzy set: Assumed Reliability (Ra), Mean (m) and Variance (v).
- Output fuzzy set: Estimated Reliability (Re).
- Types of membership function: Triangular and Trapezoid.
- Defuzzifiction method: Centroid

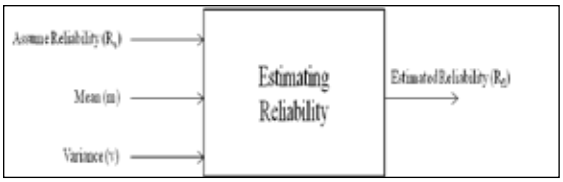

Figure 4: Fuzzy Logic model

For this first we have to define linguistic variables. We will use Assume Reliability (Ra), Mean(m) and Variance(v).

Step 1: Defines the range of linguistic variables.
Step 2: Determine Fuzzy sets Fuzzy sets can have a variety of shapes. A triangle or trapezoid is widely used.
Step 3: Construct fuzzy rules.
Step 4: Fuzzification, encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
Step 5: Using Defuzzification, evaluate and tune the system and estimate the reliability of each component.

VII. Conclusion

Reliability is one of the most important nonfunctional requirements for software. Estimating reliability of component-based software is not simple task, and it is recent research topic for many researchers to develop efficient approach for estimating reliability of component based software system. Mathematical models cannot estimate reliability accurately as it is associated with real time issues.

Fuzzy logic is a methodology to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory, we can estimate the reliability of individual component of the system and according to this individual components reliability we can estimate the reliability of the whole system. We can estimate overall reliability of system considering the contribution of a component's reliability depending upon its usage time and the path propagation probability for possible paths of execution. By optimizing the assumed reliability of individual component and according to this we can optimize the reliability of individual component and whole system.

REFERENCES

[1] Vidhu Bhardwaj, Thapar University-Patiala, "Estimating Reusability of Software Components Using Fuzzy Logic," ME Thesis July 2010. | | [2] G. Pour, "Component-Based Software DevelopmentApproach: New Opportunities and Challenges," Proceedings Technology of Object-Oriented Languages, 1998. TOOLS 26., pp. 375-383. | | [3] Musa, John. Software Reliability Engineering, New York, NY, McGraw-Hill, 1998. | | [4] J.D Musa (1987), Software Reliability measurement, prediction, application McGRAW-HILL International Edition. ISBN 0-07-100208-1 | | [5] Michael R. Lyu (May 2005) Handbook of Software Reliability Engineering: Introduction. IEEE Computer Society Press and McGraw-Hill Book Company | | [6] Swapna S. Gokhale. Kishor S. Trivedi (2002), Reliability Prediction and Sensitivity Analysis Based on Software Architecture, IEEE, Software reliability, pp 64-75, ISBN: 0-7695-1763-3. | | [7] S. Rajasekaran, "Neural Networks Fuzzy Logic and Genetic Algorithms Synthesis and Application," PHI, ISBN 81-203-2186-3. 2003. | | [8] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975. | | [9] Zadeh, L.A., "Outline of a new approach to the analysis of complex systems and decision processes," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 3, No. 1, pp. 28-44, Jan. 1973.