ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse137942020.pdf
https://doi.org/10.30534/ijatcse/2020/137942020

An Innovative Reliability Allocation Scheme for Component Based Software using Soft-Skills

Kiran Narang[1], Puneet Goswami[2]
[1]SRM University, India, kkirannarang@gmail.com
[2]SRM University, India, goswamipuneet@gmail.com

ABSTRACT

Software reliability is one of the main quality attribute of any software. It plays an important role in success of the overall software system. Unlike hardware, software does wear out due to its age or rust, but it is unreliability that makes it down, which occurs due to bugs or design faults in the software. Software reliability can be improved by adequate understanding of thoughtful software design, reliability apportionment, and thorough understanding of characteristics of software. Literature survey showed that reliability allocation in an essential task in the software design process. Increasing complexities in software and demand for bug free software has made reliability allocation a mandatory mission during design and planning phase. The component-based approach could potentially overcome difficulties associated with developing and maintaining monolithic software applications. This research proposes an approach to utilize a hardware reliability allocation technique to the reliability allocation of component based system integrating soft set skills. This method can be used to set reliability goals for individual components of component based system and respond to the query “what should be the reliability of individual components to get the target system reliability?” Sensitivity analysis has been done to compare the model with existing model. Finally, an example has been given to illustrate the effectiveness and feasibility of the proposed method.

Key words: Component based software, Fuzzy logic, linear programming, Proportionality factor, Reliability allocation, Reliability apportionment.

1. INTRODUCTION

Day by day increasing significance of software systems in critical machines and in routine life leads to the demand for quality software. Quality refers to a reliable system; it means no bug during execution period under stated conditions. It is very important to confirm the accuracy and reliability of the software; as defense, ISRO, NASA and other complex systems now usually depend on computer software. Unreliable software can lead to loss of time, money and human lives. According to IEEE reliability is defined as “The ability of a system or component to perform its required functions under stated conditions for a specified period of Time.” [1]. Substantial research has been dedicated to develop reliable and accurate software [2] [3].

One of the key techniques to achieve reliability is by reliability apportionment during early phases of development such as requirement analysis and design phase. Reliability allocation can be defined as a technique of allocating the predefined reliability objective among the subsystems. The concept of reliability allocation was initially used for hardware systems, however later it was adopted as an important aspect in software reliability as well [3] [4].

In the recent years, one of the major concerns of software developers is to accelerate the development process and to reduce overall development cost. Component Based Software Engineering (CBSE) had made it possible; it improves productivity, quality, reusability and reduces maintenance overheads. Component-based software engineering (CBSE) is an approach for software development emerged in the 1990's; that relies on the reuse of entities called 'software components'. It is a reuse-based approach of defining, implementing and composing loosely coupled independent components into systems [5].

The fundamental reliability apportionment for any system is specified by equation 1:

$$R_{complete\_system}\,(t) = \sum R_i\,(t) \qquad (1)$$

$R_{complete\_system}$ illustrates the complete system reliability, R_i (i=1, 2 ... n) is the target reliability allocated to each i_{th} subsystem at any particular time to achieve complete system reliability [6] [7].

The main contribution of the proposed model is to broaden the area of hardware reliability allocation techniques to be utilized for the all types of software systems and to the components of a component based system (CBS). This research supports software engineers to determine the reliability of each component of a system by allocating target reliability values. To calculate reliability target for each component, it is necessary to consider its behavior, performance and historical data of the system. The objective of this paper is to devise an approach that provides realistic and achievable reliability allocation values. To incorporate the component behavior, we have used major contributory factors that affect the reliability of a component and soft skill set method is used to fill up the unavailable data or lack of expert information (See section 3.3) [8][9][10][11][12]. One limitation of the proposed methodology is the assumption that all the components are connected in series architecture.

The organization of paper is as follows. Next segment considers the related work in the area of reliability allocation of systems. In section 3, discusses the three background technologies used in the proposed method i.e. traditional method of reliability allocation, fuzzy numbers and arithmetic, Soft-sets. . Section 4, discusses proposed fuzzy based reliability allocation methodology along with the results application of the proposed methodology by an example. In Section 5, comparison of proposed method is done with an existing method. Section 6 presents conclusion and future work of this research.

2. RELATED WORK

Reliability apportionment deals with setting target reliability goals for individual modules such that the users specified system reliability goal can be achieved [3]. Software reliability allocation technique gained importance in early software reliability prediction. The concept grew in the early 90s with the models given by Kubat [13], Zahedi and Asrafi [14], Lee [15], Leung [16] to mention a few. Conventional methods of reliability allocations could not incorporate knowledge and features of the components and turn out to be practically unsuitable [17] [18] [19].

Most of the scientist in the past used architecture based models for achieving target reliability of subsystems considering constraints such as budget, release period and risk etc[12] [13] [20] [21] [22]. In these papers the reliability allocation was done on the basis of one or two criterion like minimization of cost or maximization of reliability. G. S. Mahapatra and B. S. Mahapatra (2011) proposed redundancy based allocation approach with entropy as objective function. The goal of entropy based redundancy allocation problem was to find out optimal level of redundancies at each subsystem in a way that maximizes reliability. Intuitionist fuzzy optimization technique was used to analyze entropy based reliability redundancy optimization problem but the practical implementation of this research became difficult due to complex procedure [23].

In the current scenario, researchers are targeting multi-attribute based allocation methodologies by considering key influencing factors also called software characteristics for e.g. functionality, criticality, availability, complexity, reliability etc [5] [24] [25].

Yu, Zhang, Ran, Li, and Wang (2018) proposed a comprehensive reliability allocation technique. They considered failure effects, reliability costs, State of the art, working conditions and subsystem intricacy, to create a partially quantitative cost function [22]. Xu, Li, Zhou and Huang (2020) proposed a theoretical model based on importance measures (IMs) and it is competent to allocate the reliability indices to its different composed components quantitatively [10].

Catelani, Ciani, Patrizi, and Matteo Venzi (2017) discussed that allocation methods can be applied to all systems, from series architectures to redundant architectures. They also tested their method on two case studies to show the applicability of their model for complex redundant architectures. This method is based on estimated reliability, criticality and complexity etc [26].

The current research extends the work of Sriramdas, Chaturvedi, and Gargama (2014). The researchers applied fuzzy arithmetic for reliability allocation during early design and development. They described their technique with an example of transceiver [27]. They declared that their technique of hardware reliability allocation can be applied to any system; therefore the proposed research extends their technique to calculate reliability of components of CBS adjoining soft-set skills in it. Software reliability allocation problem is a decision and optimization problem. There are many factors that influence the reliability allocation [1] [2] [22] [23] [28]. Sriramdas et al. used cost, criticality, state-of-art, and complexity etc. for their research. In the proposed method, factors affecting reliability are selected by extensive literature survey and factors are increased to eight for precision in the results [29] [30] [31] [32] [33] [34] [35] [36] [37]. Operational profile and Intrinsic Redundancy are identified as important factors and included in the proposed research [31] [37] [38]. In this research, experts have used various automated tools like Sonar, Java NCSS for fetching the values for the major influencing factors selected for the research i.e. Maintenance, complexity, cost, redundancy introduction, criticality, operational profile, state of art and time of operation. However Sriramdas et al. research depends on expert knowledge for the providing the data. In addition to it, we incorporated soft- set method to fill the incomplete data (lack of information) about factors. Linear programming is used to calculate proportionate reliabilities of subsystems and to reduce the fuzziness that arises due to trapezoidal fuzzy number [27].

3. RESEARCH BACKGROUND

3.1 Traditional Method of Reliability Allocation (Factors Based)

Conventionally target system reliability (R^*) and j_{th} subsystem reliability (R_j) according to its weightage factor (w_j) is expressed by equation 2.

$$R_j = (R^*)^{wj} \tag{2}$$

Here *j*=1, 2, 3 ….n (‘j” denotes no. of Subsystems)

Weightage factor or weight (w_j) describes the importance of a particular component in component based system. Equation 3 described the formula for the calculation of $w_{j,}$ which further depends on proportionality factor Z_i [7] [9] [10].

$w_j = Z_j / \sum Z_j$ where j=1, 2, 3, ….n (3)

Proportionality factor (Z_j) depicts the relationship of component with the reliability factors selected for the apportionment [10].

In current scenario, Z_j further depends on the factors elected by the researchers for reliability allocation [14] [16] [19] [21] [22]. However these factors comprise lots of fuzzy characteristics; assigning a single numerical value may lead to imprecision and sometimes opinion from design engineer may have vagueness. Moreover, human interference in developing software causes the influencing factors to become fuzzy. Hence, there's keen requirement that software reliability allocation models must be based on the concept of fuzzy. Therefore the concept of fuzzy set theory is used in developing this model.

The membership function of a trapezoidal fuzzy number is piecewise linear and trapezoidal, which can express vagueness information caused by linguistic assessments easily and consider the characteristics of soft sets [12] [39] [40]. The idea of this manuscript is to unite the idea of trapezoidal fuzzy number for reliability allocation and soft set.

3.2 Fuzzy Trapezoidal Numbers and Arithmetic

The fuzzy trapezoidal number (TrFN) 'A' is represented by (a1, a2, a3, a4). Membership function for this fuzzy trapezoidal number is shown in Figure 1. It demonstrates the relationship of x with fuzzy set A. Arithmetic operations on two trapezoidal fuzzy numbers A and B, parameterized by (a1, a2, a3, a4) and (b1, b2, b3, b4) are described by equations (4), (5), (6) and (7) respectively [20] [23] [30] [32].

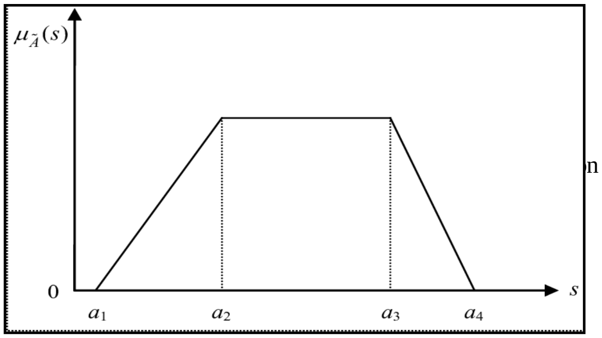


Figure 1: Fuzzy Trapezoidal Number (TrFN)

A+B = (a1+b1, a2+b2, a3+b3, a4+b4) (4)
A-B = (a1-b4, a2-b3, a3-b2, a4-b1) (5)
A*B= (a1*b1, a2*b2, a3*b3, a4*b4) (6)
A/B = (a1/b4, a2/b3, a3/b2, a4/b1) (7)

Researchers have concluded that division of fuzzy numbers escort uncertainty but it can be condensed up to a range by the use of linear programming [27] [41]. Basaran, Aladag, and Kadilar (2008) projected an investigation for division of triangular fuzzy numbers using linear programming whereas Sriramdas, Chaturvedi and Gargama (2014) proposed a method for division of fuzzy trapezoidal numbers using linear programming [27] [41]. The equation 8 shows their method of division of two trapezoidal fuzzy numbers, *X* and *Y* having parameters (*x1*, *x2*, *x3*, *x4*) and (*y1*, *y2*, *y3*, *y4*).

A=X/Y

Now A is also a trapezoidal fuzzy number parameterized by (A1, A2, A3, and A4) and equation 9, 10, 11 and 12 demonstrates the four heads of this number.

$$A_1 \geq \frac{x2}{y2} - \frac{(x2-x1)}{(y2-y1)} \quad (9) \qquad A_4 \leq \frac{x3}{g} + \frac{x4-x3}{y4-y3} \quad (11)$$

$$A_2 < \frac{x2}{y3} \quad (10) \qquad A_3 > \frac{x3}{y2} \quad (12)$$

Equation 13 shows the formulation of linear programming (LP) [41].

Max f (A) = (A4-A1) (13)

3.3 Soft Sets

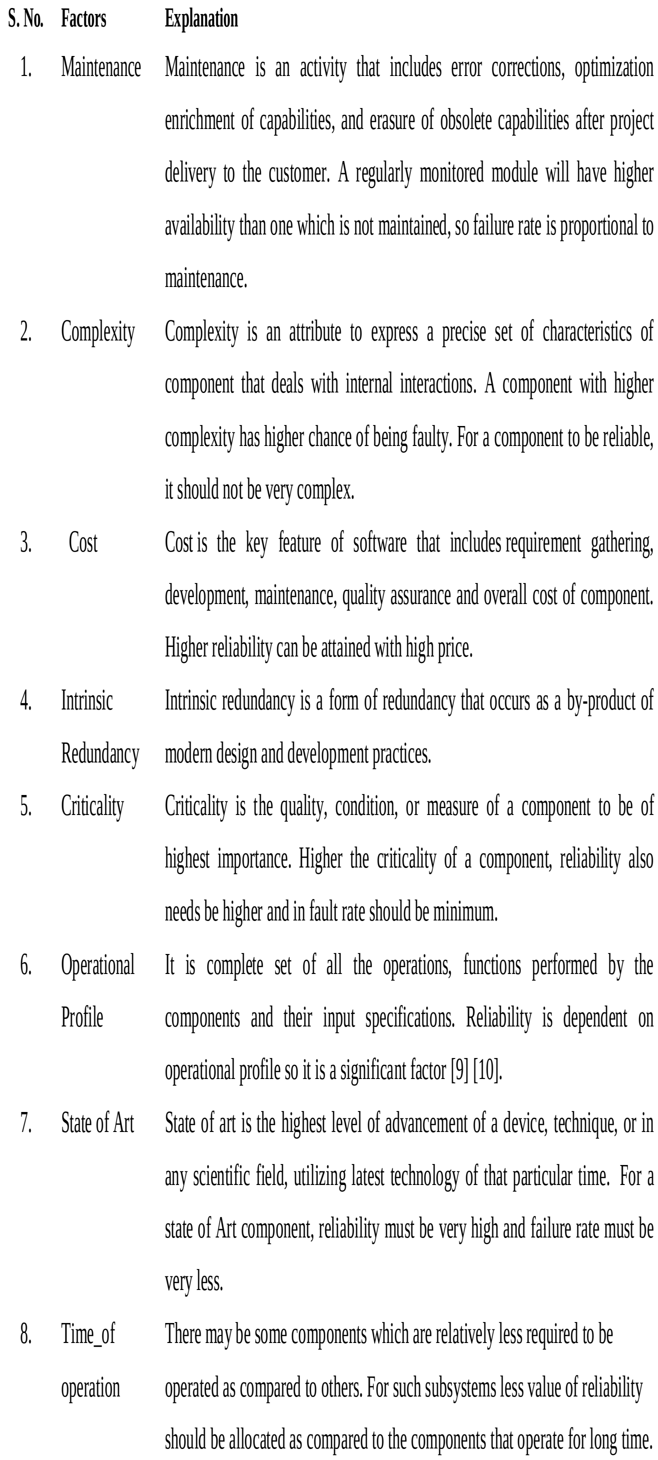
Soft set were introduced in 1999, by Molodtsov as a mathematical tool to deal with those uncertainties that cannot be handled by traditional mathematical tools. A soft set is a parameterised family of sets intuitively; this is called "soft" because the boundary of the set depends on the parameters. A soft set, over a universal set X and set of parameters E is a pair (f, A) where A is a subset of E, and f is a function from A to the power set of X. For each e in A, the set f (e) is called the value set of e in (f, A) [39].

Reliability allocation is an important activity at the design time. However, as it is done in initial phases of development, so lot of information remain unavailable during data collection. Generally this incomplete information is removed from the data set, which reduces the total number of data samples available. In this research we have applied soft set method to fill the incomplete information, based on the available data. Missing data will be filled by weighted average method, in which data available from all the experts will be aggregated to fill up the information [12] [39].

4. PROPOSED METHOD

The current research assigns target reliability to the components of Component Based System (CBS) at the stage of designing. The proposed method starts with finding major influencing factors for reliability allocation. This has been done by extensive literature survey. Table 1 describes the influencing factors selected for the current research for reliability allocation [29] [30] [31] [32] [33] [34] [35] [36] [37]. This research selects; Maintenance (M), complexity (CX), cost (C), Intrinsic Redundancy (IR), criticality (CT), operational profile (OP), state of art (S) and time of operation (T) as allocation factors for reliability allocation.

Table 1: Influencing factors Selected for the Current Research



S. No.	Factors	Explanation
1.	Maintenance	Maintenance is an activity that includes error corrections, optimization enrichment of capabilities, and erasure of obsolete capabilities after project delivery to the customer. A regularly monitored module will have higher availability than one which is not maintained, so failure rate is proportional to maintenance.
2.	Complexity	Complexity is an attribute to express a precise set of characteristics of component that deals with internal interactions. A component with higher complexity has higher chance of being faulty. For a component to be reliable, it should not be very complex.
3.	Cost	Cost is the key feature of software that includes requirement gathering, development, maintenance, quality assurance and overall cost of component. Higher reliability can be attained with high price.
4.	Intrinsic Redundancy	Intrinsic redundancy is a form of redundancy that occurs as a by-product of modern design and development practices.
5.	Criticality	Criticality is the quality, condition, or measure of a component to be of highest importance. Higher the criticality of a component, reliability also needs be higher and in fault rate should be minimum.
6.	Operational Profile	It is complete set of all the operations, functions performed by the components and their input specifications. Reliability is dependent on operational profile so it is a significant factor [9] [10].
7.	State of Art	State of art is the highest level of advancement of a device, technique, or in any scientific field, utilizing latest technology of that particular time. For a state of Art component, reliability must be very high and failure rate must be very less.
8.	Time_of operation	There may be some components which are relatively less required to be operated as compared to others. For such subsystems less value of reliability should be allocated as compared to the components that operate for long time.

Step 1: Suppose reliability is needed to be apportioned among *N* components of a CBS S_i, (i = 1 . . . *N*) and we need to allocate target reliability to these *K* components. A group of experts with *K* members will grant the fuzzy values to ith module. Rather than depending on values from single expert, data from a group of exert team is fetched and aggregated for precision. The information provided by the experts is the outcome of experiment done on various, diverse, automated, code complexity measurement tools and metrics such as TestWell CMT++, DZone Java, Sonar, JavaNCSS etc. Table 2 describes the specified scales set given to experts for evaluations of the reliability allocation factors. The grade set is described as G= {Minimum, Good, Very Good, Excellent}. Figure 2 visualizes the table 2 for trapezoidal fuzzy numbers and shows membership functions along with the scale for the allocation factors.

Data provided by all the expert members for each component about each factor is stored in the form of trapezoidal fuzzy numbers in a Table (Refer to Table 3 in example). Consequently Mij = (Mij1, Mij2, Mij3, Mij4), CXij = (CXij1, CXij2, CX_{ij3}, CX_{ij4}), C_{ij} = (C_{ij1}, C_{ij2}, C_{ij3}, C_{ij4}), IR_{ij} = (IR_{ij1}, IR_{ij2}, IR_{ij3}, IR_{ij4}), CT_{ij} = (CT_{ij1}, CT_{ij2}, CT_{ij3}, CT_{ij4}), OP_{ij} = (OP_{ij1}, OP_{ij2}, OP_{ij3}, OP_{ij4}), S_{ij} = (S_{ij1}, S_{ij2}, S_{ij3}, S_{ij4}), T_{ij} = (T_{ij1}, T_{ij2}, T_{ij3}, T_{ij4}) are the fuzzy values of *ith* component provided by *jth* expert signified as trapezoidal fuzzy number and 1, 2, 3, 4 are four heads of the trapezoidal fuzzy number.

Table 2: Fuzzy ratings/ Scales for fuzzy values from the experts

Fuzzy Values	Ratings/Scores
Excellent (E)	(7,8,9,10)
Very High (VH)	(5,6,7,8)
Good (G)	(3,4,5,6)
Minimum (M)	(1,2,3,4)

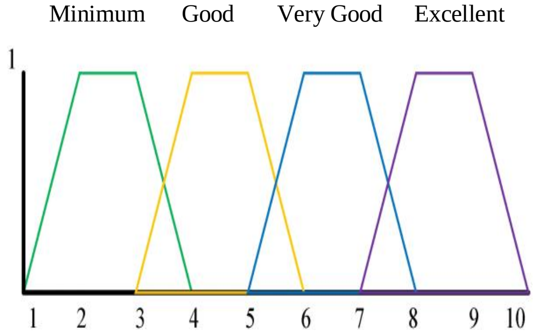


Figure 2: Fuzzy ratings for Trapezoidal Fuzzy Number

Step 2: Analyze table 2 carefully. If due to some reason, an expert is unable to provide the data for any factor of a component then soft set is used. Suppose Expert1 does not provide information about the factor "Time of Operation" (see Table 3 of example) represented by *, then the values from other experts are fetched and aggregated to fill up empty information. (See Table 4 of example)

.

Step 3: Fuzzy data provided by experts are converted to real values with help of table 2 and Figure 2. These real values are stored in the form of Trapezoidal numbers in a table (see Table 5 of example).

Step 4: Different experts have diverse observations about components, therefore it is needed to aggregate the data provided by all experts for ith component, by means of the subsequent equations 14-21 and M_i, CX_i, C_i, IR_i, CT_i, OP_i, S_i, T_i depicts the aggregation of values provided by the experts for reliability allocation factors(Refer to Table 6 in example).

Maintenance (M_i) $=\sum_{j=1}^{m} M_{ij}/m$ (14)
Complexity (CX_i) $= \sum_{j=1}^{m} CX_{ij}\ /m$ (15)
Cost (C_i)$= \sum_{j=1}^{m} C_{ij}\ /m$ (16)
Redundancy Introduction (RI_i) $= \sum_{j=1}^{m} RI_{ij}/m$ (17)
Criticality (CT_i) $= \sum_{j=1}^{m} CT_{ij}\ /m$ (18)
Operational Profile (OPi)$= \sum_{j=1}^{m} OP_{ij}\ /m$ (19)
State of art (S_i) $= \sum_{j=1}^{m} S_{ij}/m$ (20)
Time of Operation (T_i) $= \sum_{j=1}^{m} T_{ij}/m$ (21)

Step 5: Fuzzy Proportionality factor (Zj) is needed to be calculated to get the relationship of component with the reliability factors selected for the apportionment using equation 22 (see Table 7 of example).

$$Z_i = \frac{\text{Maintenance}(M) * \text{Complexity}(CX) * \text{Cost}(C) * \text{Intrinsic Redundancy (IR)}}{\text{Criticality}(CT) * \text{Operational Profile}(OP) * \text{State of art (S)} * \text{Time of operation}(T)} \quad (22)$$

Fuzzy multiplication is done using equation (6) and fuzzy division is done using the linear programming technique as described by section 3.2.

Step 6: Next step is defuzzification in which fuzzy output is translated into a single crisp value, by the degree of membership values. It is an opposite process to fuzzification, because in this, the fuzzy output is converted into crisp values. There are numerous defuzzification techniques. In our research, One of the extensively used defuzzification technique centroid method is used; that calculates center-of-gravity of the trapezoidal fuzzy number [42] [43] [44] [45] [46] [47].

Defuzzification formula for a centroid trapezoidal fuzzy number A, parameterized by (a1, a2, a3, a4) is given by equation 23 [43]. (See Table 8 of example)

$$D_a = \frac{\{(a3)^2 + (a4)^2 + a3.a4 - (a1)^2 - (a2)^2 - a1.a2\}}{\{3(a3 + a4 - a1 - a2)\}} \quad (23)$$

Step 7: Defuzzifed value D_a will assist in calculating the weightage of every component in a CBS Equation (3) (See Table 8 of example).

Step 8: For determining the target reliability of all the components of component based system, the ultimate step is to apply Equation (2) (See Table 8 of example).

4.1 Analysis of Results and description of the proposed method by an example

To show the efficiency of proposed fuzzy reliability allocation technique, we have taken an example of Enterprise JavaBeans architecture.

Enterprise JavaBeans (EJB) is a specification for developing large-scale, distributed business applications on the Java platform. The EJB architecture consists of three main components: EJB Client, the EJB Home Interface, EJB Remote Interface and Enterprise Java Beans (EJBs). EJBs run inside an EJB container, and the EJB container runs inside a Java application server.

Assume that all the components are connected in series and system's reliability to be apportioned among subcomponent is 0.875, target reliability of the components calculated by the proposed model is specified in table 7.

Step 1: We have considered a group of the three expert members (j=3), who will grant linguistic values based on their experiment to all the four components (i=4), so that target system reliability can be achieved. These experimental rating are the outcome of a variety of diverse, automated, code complexity measurement tools and metrics such as TestWell CMT++, DZone Java, Sonar, JavaNCSS etc. Table 3 shows the Fuzzy values provided by the Expert Team to all the components for the reliability allocation factors.

Table 3: Allocation of fuzzy values by the experts

COMPONENTS	EXPERT TEAM	M	CX	C	IR	CT	OP	S	T
EJB CLIENT	ET1	E	VH	E	E	G	VH	VH	VH
	ET2	VH	E	G	G	VH	G	E	VH
	ET3	VH	G	VH	*	VH	VH	VH	G
EJB HOME INTERFACE	ET1	VH	VH	E	*	VH	G	E	E
	ET2	G	VH	G	G	VH	M	E	E
	ET3	VH	G	E	E	G	VH	VH	VH
EJB REMOTE INTERFACE	ET1	vH	E	E	E	E	G	E	*
	ET2	VH	VH	E	VH	VH	VH	E	E
	ET3	E	E	VH	G	M	G	VH	E
EJB BEAN	ET1	*	G	M	E	E	G	M	E
	ET2	M	G	M	E	*	E	M	E
	ET3	VH	M	M	VH	G	VH	E	VH

*Values show missing data/unobtainable information

Step 2: Analyse the table formed in step1. Find out the places that have *. Apply softest technique to fill up all the entries. For example to fill the data for Expert1 about the factor "Time of Operation" (see Table3 of example) represented by *, values from other experts are fetched and aggregated using existing information.

Both of the experts have specified "E" for the factor "Time of Operation" for EJB Client, therefore aggregated value "E" is filled in further tables. Table 4 shows Filled information in place of * by applying soft-set method.

Step 3: Fuzzy data provided by experts is converted to real values with help of table2 and Figure 2. These real values in the form of Trapezoidal numbers are stored in table 5.

Table 4: Filling missing information by applying soft-set

COMPONENTS	EXPERT TEAM	M	CX	C	IR	CT	OP	S	T
EJB CLIENT	ET1	E	VH	E	E	G	VH	VH	VH
	ET2	VH	E	G	G	VH	G	E	VH
	ET3	VH	G	VH	VH	VH	VH	VH	G
EJB HOME INTERFACE	ET1	VH	VH	E	VH	VH	G	E	E
	ET2	G	VH	G	G	VH	M	E	E
	ET3	VH	G	E	E	G	VH	VH	VH
EJB REMOTE INTERFACE	ET1	H	E	E	E	E	G	E	E
	ET2	VH	VH	E	VH	VH	VH	E	E
	ET3	E	E	VH	G	M	G	VH	E
EJB BEAN	ET1	G	G	M	E	E	G	M	E
	ET2	M	G	M	E	VH	E	M	E
	ET3	VH	M	M	VH	G	VH	E	VH

Step 4: Different experts have diverse observation about components, therefore it is needed to aggregate the data provided by all experts for ith component, by means of the 14-21. Table 6 shows aggregated reliability allocation (fuzzy) information for four subsystems.

.

Step 5: As we have used rating of the factors, so we have to estimate fuzzy proportionality amount (Z_i) of all the subsystem. It can be done by solving equation (22), which comprises two key fuzzy functions i.e. multiplication and division. Using equation (6), Multiplication of fuzzy data can be performed where as fuzzy division operation is executed by linear programming method. Table 7 depicts the fuzzy proportionality factor for each subsystem.

Step 6: To discover the weightage of all components of a component based system, we need to defuzzify the proportionality factor by centroid trapezoidal formula given in equation 23. Table 8 depicts the defuzzified values of all subsystems.

Step 7: Now using equation (2) we will calculate the weightage of all the Subsystems. Table 8 depicts the weightage of all subsystems.

Step 8: Last and final step is to find the target reliability, which is gained by equation (3) for each subsystem assuming reliability of system to be .875. Table 8 shows reliability allocated to every component by the proposed model. Figure 3 describes the graphical Analysis of the results. It demonstrates that the reduced defuzzified value leads to higher reliability allocation for a particular sub module. In this way we can apply the proposed methodology to calculate the reliability allocation for a component based system.

Table 5: Conversion of values into fuzzy scale

COMPONENTS	EXPERT TEAM	M	CX	C	IR	CT	OP	S	T
EJB CLIENT	ET1	(7,8,9,10)	(5,6,7,8)	(7,8,9,10)	(7,8,9,10)	(3,4,5,6)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)
	ET2	(5,6,7,8)	(7,8,9,10)	(3,4,5,6)	(3,4,5,6)	(5,6,7,8)	(3,4,5,6)	(7,8,9,10)	(5,6,7,8)
	ET3	(5,6,7,8)	(3,4,5,6)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)	(3,4,5,6)
EJB HOME INTERFACE	ET1	(5,6,7,8)	(5,6,7,8)	(7,8,9,10)	(5,6,7,8)	(5,6,7,8)	(3,4,5,6)	(7,8,9,10)	(7,8,9,10)
	ET2	(3,4,5,6)	(5,6,7,8)	(3,4,5,6)	(3,4,5,6)	(5,6,7,8)	(1,2,3,4)	(7,8,9,10)	(7,8,9,10)
	ET3	(5,6,7,8)	(3,4,5,6)	(7,8,9,10)	(7,8,9,10)	(3,4,5,6)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)
EJB REMOTE INTERFACE	ET1	(5,6,7,8)	(7,8,9,10)	(7,8,9,10)	(7,8,9,10)	(7,8,9,10)	(3,4,5,6)	(7,8,9,10)	(7,8,9,10)
	ET2	(5,6,7,8)	(5,6,7,8)	(7,8,9,10)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)	(7,8,9,10)	(7,8,9,10)
	ET3	(7,8,9,10)	(7,8,9,10)	(5,6,7,8)	(3,4,5,6)	(3,4,5,6)	(3,4,5,6)	(5,6,7,8)	(7,8,9,10)
EJB BEAN	ET1	(3,4,5,6)	(3,4,5,6)	*(1,2,3,4)	(7,8,9,10)	(7,8,9,10)	(3,4,5,6)	(1,2,3,4)	(7,8,9,10)
	ET2	(1,2,3,4)	(3,4,5,6)	(1,2,3,4)	(7,8,9,10)	(5,6,7,8)	(7,8,9,10)	(1,2,3,4)	(7,8,9,10)
	ET3	(5,6,7,8)	(1,2,3,4)	(1,2,3,4)	(5,6,7,8)	(3,4,5,6)	(5,6,7,8)	(7,8,9,10)	(5,6,7,8)

Table 6: Cumulative reliability allocation (fuzzy) information for four subsystems

COMPONENTS	M	X	C	IR	CT	OP	S	T
EJB CLIENT	(5.6,6.6,7.6,8.6)	(5,6,7,8)	(5,6,7,8)	(5,6,7,8)	(4,3,5.3,6.3,7.3)	(4,3,5.3,6.3,7.3)	(5.6,6.6,7.6,8.6)	(4,3,5.3,6.3,7.3)
EJB HOME INTERFACE	(4,3,5.3,6.3,7.3)	(4,3,5.3,6.3,7.3)	(5.6,6.6,7.6,8.6)	(5,6,7,8)	(4,3,5.3,6.3,7.3)	(3,4,5,6)	(6.3,7.3,8.3,9.3)	(6.3,7.3,8.3,9.3)
EJB REMOTE INTERFACE	(5.6,6.6,7.6,8.6)	(6.3,7.3,8.3,9.3)	(6.3,7.3,8.3,9.3)	(5,6,7,8)	(5,6,7,8)	(3.6,4.6,5.6,6.6)	(6.3,7.3,8.3,9.3)	(7,8,9,10)
EJB BEAN	(3,4,5,6)	(2.3,3.3,4.3,5.3)	(1,2,3,4)	(6.3,7.3,8.3,9.3)	(5,6,7,8)	(5,6,7,8)	(1.6,2.6,3.6,4.6)	(6.3,7.3,8.3,9.3)

.

Table 7: Estimation of the reliability proportionality factor for five subsystems

Components	M*X*C*IR	CT*OP*S*T	Proportionality Factor (Z_I)
EJB Client	(700,1425.6,2606.8,4403.2)	(445.23,982.58,1900.35,3775.19)	(.10,1.276, 2.32, 2.653)
EJB Home Interface	(517.72,1112.36,2111.50,3666.35)	(512,1129.74,2170.03,3788.26)	(.017,.512,1.869,1.933)
EJB Remote interface	(1111.30,2110.28,3664.94,5950.51)	(793.8,1611.84,2928.24,4910.4)	(.72,1.31,2.27,2.40)
EJB Bean	(43.47,192.72,535.35,1182.96)	(252,683.28,1464.12,2737.92)	(0, .131, .783, .873)

Table 8: Final Reliability Apportionment of Every Module in the System

Components	Defuzzifed Values	Weightag e	Reliability Allocated
EJB Client	1.5758	.338	0.968
EJB Home Interface	.9618	.206	0.974
EJB Remote Interface	1.664	.357	0.967
EJB Bean	.447	.0961	0.987

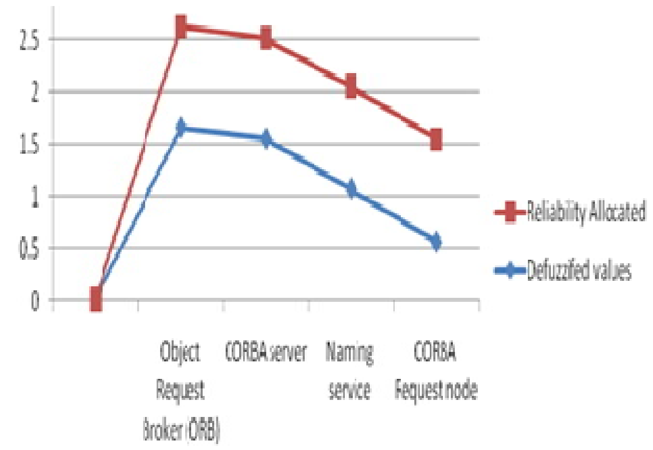


Figure 3: Analysis of Results

5. COMPARISON AND SENSITIVITY ANALYSIS

The result depicts that hard ware reliability allocation method proposed by Sriramdas et al. can be successfully utilized for the components of component based software.

Enhancement in the proposed model is achieved in certain ways and shown by figure 4. Accurate results are obtained as all the experts have used diverse automated tools to fetch the data required by the model; however, Sriramdas et al. research depends on the expertise of team member that may not be reliable sometimes. To obtain precise result eight factors are utilized instead of six. Current research is proficient to use soft skill method to fill aggregated data in case it is not available for any factor due to any reason. Figure 5 visualizes the comparative analysis of both the models. Precision and accuracy is achieved by selecting extremely significant factors for the

reliability as compared to traditional models that does not use factor based allocation.

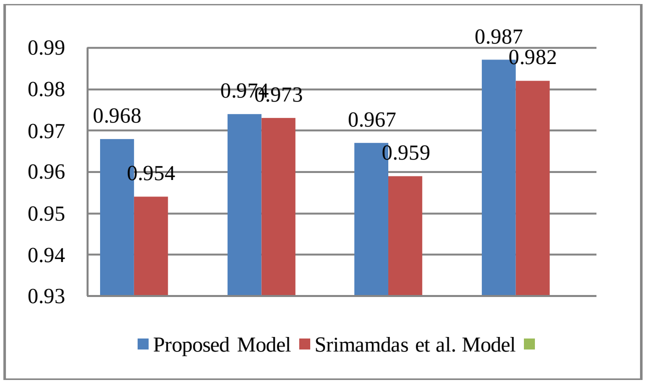


Figure 5: visualizes the comparative analysis of the models

6. CONCLUSION AND FUTURE WORK

The proposed method is generalized research that can be applied to any software such as component based software or even to any hardware appliance. It is a weightage based reliability allocation method, so weightage of each component is calculated, based on proportionality factor and finally the target reliability is allocated to each module.

The major benefit of current reliability allocation method is its flexibility, to be applied on any hardware or software system. Application area of this methodology covers a vast domain to include any electrical or mechanical device. This research work extends the scope of all the hardware reliability apportionment techniques to be applicable on new and complex softwares.

Linear programming technique is used so that fuzziness in trapezoidal fuzzy division can be reduced. For defuzzification centroid method is used. An expert panel provides the data fetched from automated tools for the various reliability factors of the components.

Sensitivity analysis has been performed to demonstrate the performance of proposed approach. The model presented in this paper guarantee about the apportionment of software system reliability among its components.

This technique can be further improved by implementing it for parallel software architectures and mixed architectures. In addition to it fuzziness that arises due to the use of trapezoidal fuzzy numbers can further be reduced by coming technologies. A new concept of relative importance of factors, according to the nature of system can also be collaborated.

REFERENCES

1. K. K. Aggarwal. ***Reliability Engineering***, 3rd ed. Netherlands: Kluwar Academic Publisher, 2007, pp. 20-52.
2. C. E. Ebeling**.** ***An Introduction to Reliability and Maintainability Engineering,*** 10th ed. New Delhi India: Tata Mc- Graw Hill Publishing Company Limited, 2008, pp. 115-138.
3. K. K. Aggarwal, and Y. Singh. **Software reliability apportionment using analytical hierarchy process,** *ACM SIGSOFT Software Engineering Notes,* Vol. 20, no. 5, pp. 56–61, 1995. https://doi.org/10.1145/217030.217041
4. B. S. Dhillon. **Reliability apportionment/ allocation: A survey,** *Microelectronics Reliability*, Vol. 26, no. 6, pp. 1121–1129, 1986.
5. K. Narang and P. Goswami. **Comparative Analysis of Component Based Software Engineering Metrics**, *International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 1-6 (2018).
6. M. Garg and R. Lai. **A Method for Selecting a Model to Estimate Software Reliability at the Design Phase of Component-Based Real Time System Development**, *Journal of Software*, Vol. 13, no. 6, pp. 317-334, June 2018. https://doi.org/10.17706/jsw.13.6.317-334
7. Z. Hong-bin and J. Zhi-xin. **Complex system reliability allocation based on fuzzy decision method**, *Proceedings of the IEEE international workshop on intelligent systems and applications*, pp. 1–4, 2009.
8. G. S. Mahapatra and B.S. Mahapatra**. Redundancy optimization using intuitionist fuzzy multi-objective programming,** *International Journal of Performability engineering,* Vol. 7, no. 2, pp. 155–164, 2011.
9. G. Yingkui and T. Shuyun. **A fuzzy reliability allocation method for the product based on the knowledge**, *Proceedings of the IEEE International Conference on Measuring Technology and Mechatronics Automation*, 2010.
10. M. Xu, D. Li, S. Q. Zhou, and XJ. Huang. **Reliability Allocation Based on Importance Measures Innovative Technologies for Instrumentation and Control Systems**. *SICPNPP 2019. Lecture Notes in Electrical Engineering*, Vol. 595, Singapore 2020.
11. K. H. Chang. **A novel reliability allocation approach using the OWA tree and soft set**, *Reliability Management and Computing, Springer Science & Business Media,* New York, 2016.
12. Z. W Li, and T. S. Xie. **The relationship among soft sets, soft rough sets and topologies**, *Soft Computing*, vol. 18, no. 4, pp. 717–728, 2014.
13. P. Kubat. **Assessing reliability of modular software**, *Operations Research Letters,* vol. 8, pp. 35 – 41, 1989.
14. F. Zahedi and N. Ashrafi. **Software reliability allocation based on structure, utility, price, and cost,** *IEEE Transactions on Software Engineering*, vol. 17, no. 4, pp. 345-356, 1991.
15. H. Lee, "A structured methodology for software development effort prediction using the analytic hierarchy process," Journal of Systems and Software, vol. 21.2, pp. 179-186, 1993.
16. Y. W. Leung. **Software reliability allocation under an uncertain operational profile**, *Journal of the Operational Research Society*, vol. 48, no.4, pp. 401-411, 1997.
17. M. Kumar and K. Seetharam. **Reliability prediction using fuzzy logic**, *International Conference on Computing for Sustainable Global Development*, 2014.
18. K. Tyagi and A. Sharma. **A rule-based approach for estimating the reliability of component-based systems**, *Advances in Engineering Software*, 2012.
19. Y. W. Leung. **Software reliability allocation under an uncertain operational profile**, *Journal of the Operational Research Society*, vol. 48, no. 4, pp. 401-411, 1997. https://doi.org/10.1038/sj.jors.2600384
20. H. M. Lee. **A new algorithm for applying fuzzy set theory to evaluate the rate of aggregative risk in software development,** *Information Sciences*. Vol. 79, no. 3, pp. 323–336, 2003.
21. O. Golbasi and N. Demirel. **Risk-based reliability allocation methodology to set a maintenance priority among system components: A case study in mining**, *Maintenance and Reliability,* Vol. 19, no. 2, pp. 191–202, 2017.
22. H. Yu, G. Zhang, Y. Ran, M. Li and Y. Wang. **A comprehensive and practical reliability allocation method considering failure effects and reliability costs**, *Maintenance and Reliability,* Vol. 20, no. 2, pp. 244–25, 2018.
23. G. S. Mahapatra and B. S Mahapatra. **Redundancy optimization using intuitionistic fuzzy multi-objective programming,** *International Journal of Performability Engineering,* Vol. 7, no. 2, pp. 155–164, 2011.
24. Sedigh Ali, S. Gafoor, A. Paul, A. Raymond. **Software Engineering Metrics for COTS based Systems**, *IEEE Computer*, pp 44-50, May 2001.

25. Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki, Fukazawa**. A Metrics Suite for Measuring Reusability of Software Components,** *9th IEEE International Software Metrics Symposium (METRICS 2003)*, Sydney, Australia. IEEE Computer Society 4, No. 8, pp. 57-61, 2013.
26. M. Catelani, L. Ciani, G. Patrizi, and M. Venzi**. Reliability Allocation Procedures in Complex Redundant Systems,** *IEEE Systems Journal*, pp. 1-11, 2017.
27. V. Sriramdas, S. K. Chaturvedi, and H. Gargama. **Fuzzy arithmetic based reliability allocation approach during early design and development,** *Expert Systems and Applications*, Vol. 41, no. 7, pp. 3444–3449, 2014. https://doi.org/10.1016/j.eswa.2013.10.048
28. Amrita and D. K. Yadav. **Development of Software Operational Profile**, *International Journal of Applied Engineering Research*, Vol. 12, no. 22, pp.11865-11873, 2017.
29. K. Narang and P. Goswami. **DRCE Maintainability Model for Component Based Systems using Soft Computing Techniques**, *International Journal of Innovative Technology and Exploring Engineering,* vol. 8, no. 9, pp. 2552-2560, (2019).
30. K. Narang, P. Goswami and R. K. Kumar. **Maintainability Configuration for Component Based Systems using Fuzzy Approach**, *International Conference on Computational Methods and Data Engineering*, 2020.
31. D. K. A. Yadav. **Operational Profile Based Software Test Case Allocation**, *Proceedings of 2nd International Conference on Computing for Sustainable Development*; pp. 1775-1779, 2015.
32. H. Gargama and S. K. Chaturvedi. **Criticality assessment models for failure mode effects and criticality analysis using fuzzy logic**, *IEEE Transactions on Reliability*, Vol. 60, no. 1, pp.102–108, 2011.
33. M. Molani, A. Ghaffari, and A Jafarian. **A new approach to software project cost estimation using a hybrid model of radial basis function neural network and Genetic Algorithm**, *Indian Journal of Science and Technology*, Vol. 7, no. 6, pp. 838–843, 2014.
34. G. Gayathry and R. T. Selvi. **Classification of software reliability models to improve the reliability of software**, *Indian Journal of Science and Technology*, Vol. 8, no. 29, pp. 1–5, 2015.
35. A. O. C. Elegbede. **Reliability allocation through cost minimization**, *IEEE Transactions on Reliability*, Vol. 52, no. 1, pp. 106-111, 2003. https://doi.org/10.1109/TR.2002.807242
36. Li. Xiang, C. Mutha, and C. S. Smidts. **An automated software reliability prediction system for safety critical software,** *Empirical Software Engineering*, 2015.
37. J. D. Musa. **Operational Profiles in Software Reliability Engineering**, *AT&T Bell Laboratories,* pp.14-32, 1993.
38. P. K. Chaurasia and J. Ali**. Operational Profile: A Critical Review**, *Journal of Global Research in Computer Science*, Vol. 4, no. 8, pp. 57-61, 2013.
39. D. A Molodtsov. **A theory of soft sets**. *Moscow: Editorial URSS*, 2004.
40. D. Molodtsov. **Soft set theory first results**, *Computers and Mathematics with Applications*, Vol. 37, no. (4–5), pp. 19–31, 1999.
41. M. A. Basaran, C. H. Aladag and C. Kadilar. **Conducting fuzzy division by using linear programming**, *WSEAS Transactions on Information Science and Applications,* Vol. 5, no. 6, pp.923–928, 2008.
42. R. E. Giachetti and R. E. Youn. **Analysis of the error in the standard approximation used for multiplication of triangular and trapezoidal fuzzy numbers and the development of a new approximation**, *Fuzzy Sets and Systems,* Vol. 91, no. 1, pp. 1–13, 1997.
43. Y. M. Wang, J. B. Yanga, D. L. Xua, and K. S. Chin. **On the Centroid of fuzzy numbers**, *Fuzzy Sets and Systems*, Vol. 157, No. 7, pp. 919– 926, 2006. https://doi.org/10.1016/j.fss.2005.11.006
44. L. A. Zadeh. **Fuzzy sets**, *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.
45. Amrita and D. Yadav. **Software Reliability Apportionment using Fuzzy Logic**, *Indian Journal of Science and Technology,* Vol. 9, No. 45, December 2016.
46. M. H. Molawad and S. D. Joshi. **Software reliability prediction using Knowledge Engineering approach,** *International Journal of Advanced Trends in Computer Science and Engineering,* Vol. 8, no.6,pp. 2768-2772,2019. https://doi.org/10.30534/ijatcse/2019/14862019
47. R. Marco, N S. Herman, and S. S. S. Ahmad. **Optimizing Software Effort Estimation Models Based On Metaheuristic Methods: A Proposed Framework,** *International Journal of Advanced Trends in Computer Science and Engineering,* Vol. 8, no.1.5, pp.294-304, 2019. https://doi.org/10.30534/ijatcse/2019/5181.52019