

# **Assessment of Architecture-based Software System Reliability Allocation on Components using a Dynamic Programming**

**Srinivasarao.Sabbineni**  
Assistant Professor,  
Computer Science and  
Engineering Department,  
K.L.University,Vaddeswaram,  
Guntur,Andhrapradesh State -  
522502, India

**Kurra.Rajasekhara  
Rao,PhD**  
Professor,  
Computer Science and  
Engineering Department,  
K.L.University,Vaddeswaram,  
Guntur,Andhrapradesh State -  
522502, India

## **ABSTRACT**

In this paper, Software system reliability allocation during the software product design phase of SDLC .A system is made up several elements or components in simple or complex systems. We used Architecture-based approach for modeling software reliability optimization problem, on this basis a dynamic programming has been used to allocate the reliability to each component so as to minimize the cost of designing phase of software while meeting the chosen reliability. The result of our research show an optimal solution or near optimal to the problem of choosing the component containing the software can be achieved with lower cost.

## **Keywords**

Software reliability, Reliability Allocation, Reliability Estimation, Software model, Dynamic Programming, Architecture .

## **1. INTRODUCTION**

The Architecture-based Software system Reliability is the probability that a system or component can perform its intended function for a specified interval under stated conditions. The complement of reliability is the probability of failure.

Reliability is one of the main essential and quality types of software since it events software failures in the development process. Complex level reliability of software can absolutely result in extra development cost. In practical engineering, software project managers are necessary to estimate the cost needed to complete the development of software system pleasing into account the required level of reliability. Indeed, the cost should be estimated ,as early and perfectly as possible, so that the development team can choose the good techniques in order to address constraints of cost and essential reliability.

Several software reliability models had been proposed in the earlier decades to help software managers and designers to analyze and design systems under such dual and often conflicting constraints of maximizing reliability and minimizing cost. In this paper we clarified design phase of each component reliability allocation in small and large case software. In SDLC for design phase we know the main limitations. First one is system size and second one is project structure. I am taking first one is system size. System size means number of components .The components may be depends on the size of the project structure.

The discussion of this paper takes the following structure. Section 2 introduces Architecture-based software reliability allocation model. Section 3 depicts how to find out the optimal allocation method by using a dynamic programming algorithm. Section 4 illustrates the application of the algorithm proposed in section 3 and section 5 offers conclusion and future work are presented.

## **2. ARCHITECTURE-BASED SOFTWARE RELIABILITY ALLOCATION MODEL**

### **2.1 Software system development cost minimization versus reliability allocation**

In fact, it is difficult to develop the software reliability while releasing the software system development cost because they are two inconsistent constraints. The invented software development cost minimization can be measured from two facts of assessments. One is to find an optimal reliability allocation method while achieving the given reliability such that the development cost can be as small as possible; the other is how to allocate the reliability to each component on the basis of the given cost so that the system reliability can be maximized. This paper focuses on the former one.

Several systems are executed by using a set of interconnected subsystems. Reliability allocation means fixing the reliability among different subsystems so that the total system development cost (including human, material resources, development time and testing time etc) can be minimized. Reliability allocation can be used to planning with such kind of problem that the goal is set prior to the solution. Normally the number of the solution is more than one, as a result reliability allocation is used to deal with the optimal problem with some restrictions.

## 2.2 Software reliability and cost model

It is reasonable to assume that the cost function  $f_i$  would satisfy these three conditions.

- $f_i$  is a positive definite function
- $f_i$  is non-decreasing.
- $f_i$  increases at a higher rate for higher values of  $R_i$

The third condition suggests that it can be very expensive to achieve the reliability value of 1.

According to the definition of software reliability;

$$r(t) = e^{-\lambda t} \text{-----(1)}$$

where  $r(t)$  is continuous-time system reliability,  $\lambda$  is its failure rate.

According to the experience from practical engineering, this paper takes account of the factors by assuming that software reliability and development cost Satisfithe relations as below:

1).Software development cost is inversely proportional to the number of system failures.

$$E \propto 1/C \text{-----(2)}$$

$$\lambda \propto E \text{-----(3)}$$

where  $C$  is the development cost and  $E$  is the number of system failures.using equation(2),reliability function (1) is therefore

$$C \propto -1/\ln r \text{-----(4)}$$

2.software development cost is proportional to the complexity and size of the software system.

Taking assumption 2) into account,the equation (4) can be expressed as:

$$C = -\alpha/\ln r \text{-----(5)}$$

Where  $\alpha$  represents the complexity size for developing the software system and is called reliability cost coefficient in this paper.

In consideration of the operation in practical engineering ,where personal training ,development tools preparation etc. we use character  $\beta$  to represent the basic development cost in this paper.

Reliability cost model as below:

$$C(r) = -\alpha/\ln r + \beta \text{-----(6)}$$

Equation(6) states the relationship between system development cost and system reliability in our model.

Given a system with many components,the reliability of its component  $I$  can be also stated as:

$$C(R_i) = \alpha_i/\ln R_i + \beta_i \text{-----(7)}$$

## 2.3 Software reliability model

Here we assume the software system has been designed as an assembly of appropriated connected components. Let there be  $n$  components, each with reliability  $R_i$  and cost  $C_i, i=1,2,...,n$ .let  $R_{obj}$  be the specified target reliability and the total system cost.Let  $F(R_1, R_2, \dots, R_m)$  be the function of  $R$  and  $R_i$ .

The software reliability allocation model can be stated as:  
Objective function

$$\text{Min } C \text{-----(8)}$$

Subject to

$$\{ r = F(r_1, r_2, \dots, r_n) \geq R_{obj}, 0 < r_i < 1 \text{-----(9)}$$

$$C = \sum C_i = \sum \beta_i - \sum \alpha_i / \ln r_i \text{-----(10)}$$

If  $\beta$  is simplified to 0(zero).then eq(10) can be rewriting as:

$$C = \sum C_i = -\sum \alpha_i / \ln r_i \text{-----(11)}$$

As for a software system with  $n$  sub systems or components within a given architecture ,the development system will increase while enhancing the software reliability target. Significantly ,the greater improvement of the reliability ,the more increase of the cost will be required.

Such a problem can be resolved with variety methods,and dynamic programming is an option.

## 3. EXPENDING DYNAMIC PROGRAMMING ALGORITHM TO SOLVE RELIABILITY ALLOCATION PROBLEM:

A software system with  $n$  components and the association function  $F$  discussed above is known. The reliability-cost coefficient  $\alpha$  of each component and the specified system reliability target  $R_{obj}$  is given.

The dynamic programming algorithm is as follows:

Step 1: Let  $S$  represent the reliability matrix  $[r_1, r_2, \dots, r_n]$ ,  $T$  represent the cost matrix  $[c_1, c_2, \dots, c_n]$ ,  $\delta$  be the solving step length,  $I_i$  represent the matrix with one column and  $n$  rows in which only the value of the  $i$ th element is 1 and the rest are all 0. Assume  $S_0 = [\max_r, \max_r, \dots, \max_r]$ ,  $\max_r$  represents the maximized possible reliability, for example 0.9999, which means the initial reliability values of the components are all

maxr.

Step 2: As for  $S_0, T_0$  can be solved(11),  $C_0$  can be given by and system reliability  $R_0$  can be given by function F.

Step 3: If  $R_0 < R_{obj}$  then stop and return. No solutions.

Step 4: Set Rate=0;

Step 5: for i=1 to n

- i)  $S' = S_0 - I_i * \delta$ ;
- ii) With regard to  $S'$ , Generate reliability  $R'$  with the function F,  $T'$  with (7), total cost  $C'$
- iii)  $\Delta C = C_0 - C'$ ;  $\Delta R = R_0 - R'$ ;

iv) if  $R' \geq R_{obj}$  and  $\Delta C / \Delta R > Rate$  then Set Rate= $\Delta C / \Delta R$ ,  $R = R'$ ,  $S = S'$ ,  $C = C'$ ,  $T = T'$ ;

Step 6: if  $R_0 \neq R$  then set  $S_0 = S$ ;  $R_0 = R$ ;  $C_0 = C$ ;  $T_0 = T$ ;

return to step 4

Where reliability allocation result  $S_0$  is the reliability of each component.  $R_0$  and  $C_0$  are the corresponding system reliability and expected system development cost.  $T_0$  is the expected development cost allocated to each component. Notice from the above that prerequisite to the correctness of the algorithm is that the decrease in reliability of one component can result in that of the whole system and lower the development cost. But that can be guaranteed in our algorithm. The aim of step 5 iv) in the above algorithm is to select an optimal component whose decrease in reliability can result in the maximal cost/reliability variation, which makes the single step programming optimized so that optimal reliability allocation of the ultimate system is guaranteed.

#### 4. Example and Result

Here we choose a system with four independent components  $r_1, r_2, r_3$ . We assume that all the components are essential to the system and their failures are statistically independent. Therefore, the relationship between the total system reliability  $r$  and its components' reliability  $r_i$  ( $i=1, 2, 3$ ) can be stated as:  $r = F(r_1, r_2, r_3) = r_1 * r_2 * r_3$  Suppose that the complexities of the components are 0.45, 0.82 and 0.64 respectively. In order to minimize the system development cost and the system reliability shall be no less than 0.95, how to allocate the reliability to each component. Set the precision of computing is 0.01.

Such a problem can be rewritten as:

$$R = r_1 * r_2 * r_3 \leq 0.95$$

$$C_1 = -0.45 / \ln r_1$$

$$C_2 = -0.82 / \ln r_2$$

$$C_3 = -0.64 / \ln r_3$$

Compute the values of parameters  $(r_1, r_2, r_3)$  with which the total cost  $C$  ( $C = c_1 + c_2 + c_3$ ) is minimized. With respect to each component, we compute the cost with the reliability from 0.95 to 0.99 (increment is 0.01) according to the reliability/cost function model in the data set as shown in Table 1.

**TABLE 1: COST AND RELIABILITY DATASET**

s.no	r1	c1	r2	c2	r3	c3
1	0.95	8.77	0.95	15.98	0.95	12.47
2	0.96	11.02	0.96	20.08	0.96	15.67
3	0.97	14.77	0.97	26.92	0.97	21.01
4	0.98	22.27	0.98	40.58	0.98	31.67
5	0.99	44.77	0.99	81.58	0.99	63.67

According to the algorithm above, set initial state  $S_0 = [0.99, 0.99, 0.99]$ . Accordingly,  $T_0 = [44.77, 81.58, 63.67]$ ,  $\delta = 0.01$ , and the system cost  $C_0 = 44.77 + 81.58 + 63.67 = 190.02$ , system reliability  $R_0 = 0.99 * 0.99 * 0.99 = 0.97$ .

Set  $i=1, 2, 3$  then compute separately with different value:

1)  $S' = S_0 - [0.01, 0, 0] = [0.98, 0.99, 0.99]$ ,  $R' = 0.96$ ,  $T' = [22.27, 81.58, 63.67]$ ,  $C' = 167.52$   $\Delta C = 22.5$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 2250$

2)  $S' = S_0 - [0, 0.01, 0] = [0.99, 0.98, 0.99]$ ,  $R' = 0.96$ ,  $T' = [44.77, 40.58, 63.67]$ ,  $C' = 149.02$   $\Delta C = 41$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 4100$

3)  $S' = S_0 - [0, 0, 0.01] = [0.99, 0.99, 0.98]$ ,  $R' = 0.96$ ,  $T' = [44.77, 81.58, 31.67]$ ,  $C' = 158.02$   $\Delta C = 32$ ;  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 3200$

Choose the optimal result 2), set  $S_0 = [0.99, 0.98, 0.99]$ , continue to perform the same operation :

1)  $S' = S_0 - [0.01, 0, 0] = [0.98, 0.98, 0.99]$ ,  $R' = 0.95$ ,  $T' = [22.27, 40.58, 63.67]$ ,  $C' = 126.52$ ,  $\Delta C = 22.5$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 2250$

2)  $S' = S_0 - [0, 0.01, 0] = [0.99, 0.97, 0.99]$ ,  $R' = 0.95$ ,  $T' = [44.77, 26.92, 63.67]$ ,  $C' = 135.36$   $\Delta C = 13.66$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 1366$

3)  $S' = S_0 - [0, 0, 0.01] = [0.99, 0.98, 0.98]$ ,  $R' = 0.95$ ,  $T' = [44.77, 40.58, 31.67]$ ,  $C' = 117.02$   $\Delta C = 32$ ;  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 3200$

Choose the optimal result 3), set  $S_0 = [0.98, 0.99, 0.99]$ , continue to perform the same operation :

1)  $S' = S_0 - [0.01, 0, 0] = [0.97, 0.99, 0.99]$ ,  $R' = 0.95$ ,  $T' = [14.77, 81.58, 63.67]$ ,  $C' = 160.02$ ,  $\Delta C = 7.5$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 750$

2)  $S' = S_0 - [0, 0.01, 0] = [0.98, 0.98, 0.99]$ ,  $R' = 0.95$ ,  $T' = [22.27, 40.58, 63.67]$ ,  $C' = 126.52$   $\Delta C = 41$ ,  $\Delta R = 0.01$ ,  $\Delta C / \Delta R = 4100$

3)  $S' = S_0 - [0, 0, 0.01] = [0.98, 0.99, 0.98]$ ,  $R' = 0.95$ ,  $T' = [22.27, 81.58, 31.67]$ ,  $C' = 135.52$   $\Delta C = 32$ ;  $\Delta R = 0.01$ ,  $\Delta C/\Delta R = 3200$

Choose the optimal result 4), set  $S_0 = [0.99, 0.99, 0.98]$ , continue to perform the same operation :

1)  $S' = S_0 - [0.01, 0, 0] = [0.98, 0.99, 0.98]$ ,  $R' = 0.95$ ,  $T' = [44.77, 40.58, 31.67]$ ,  $C' = 117.02$ ,  $\Delta C = 41$ ,  $\Delta R = 0.01$ ,  $\Delta C/\Delta R = 4100$

2)  $S' = S_0 - [0, 0.01, 0] = [0.99, 0.98, 0.98]$ ,  $R' = 0.94$ ,  $T' = [34.82, 25.74, 73.63, 45.04]$ ,  $C' = 179.23$   $\Delta C = 71.5$ ,  $\Delta R = 0.01$ ,  $\Delta C/\Delta R = 7150$

3)  $S' = S_0 - [0, 0, 0.01] = [0.99, 0.99, 0.97]$ ,  $R' = 0.95$ ,  $T' = [44.77, 81.58, 21.01]$ ,  $C' = 147.36$   $\Delta C = 10.66$ ;  $\Delta R = 0.01$ ,  $\Delta C/\Delta R = 1066$

all of the results  $R'$  are less than the specified reliability target 0.95. Therefore, the reliability allocation in all cases is as below:

CASE (i):

1) System reliability allocation  $S_0 = [0.99, 0.99, 0.99]$ ;

2) System reliability  $R_0 = 0.96$ ;

3) Expected system development cost  $C_0 = 190.02$ ;

4) Expected development cost assigned to each components  $T_0 = [44.77, 81.58, 63.67]$ .

CASE (ii):

1) System reliability allocation  $S_0 = [0.99, 0.98, 0.99]$ ;

2) System reliability  $R_0 = 0.96$ ;

3) Expected system development cost  $C_0 = 149.02$ ;

4) Expected development cost assigned to each components  $T_0 = [44.77, 40.58, 63.67]$ .

CASE (iii):

1) System reliability allocation  $S_0 = [0.98, 0.99, 0.99]$ ;

2) System reliability  $R_0 = 0.96$ ;

3) Expected system development cost  $C_0 = 167.52$ ;

4) Expected development cost assigned to each components  $T_0 = [22.27, 81.58, 63.67]$ .

CASE (iv):

1) System reliability allocation  $S_0 = [0.99, 0.99, 0.98]$ ;

2) System reliability  $R_0 = 0.96$ ;

3) Expected system development cost  $C_0 = 158.02$ ;

4) Expected development cost assigned to each components  $T_0 = [44.77, 81.58, 31.67]$ .

## 5. CONCLUSION

Software reliability allocation for system components during software product design phase and implementation phase which has close relationship with software modeling and cost evaluation. We formulated an architecture-based approach for displaying software reliability optimization problem, on this basis a dynamic programming algorithm has been supported in this paper which can be used to allocate the reliability to each component so as to minimize the cost of designing software system components while meeting the desired reliability independent. The result of our experiment show an optimal or approximate optimal solution to the problem of selecting the various sizes of components comprising software can be obtained with lower cost (a high reliability). The reliability and cost allocation model presented in this paper can be used to solve the optimal allocation problems in simple and complex systems

## 6. ACKNOWLEDGEMENTS

I am thankful to Dr.K.Rajasekhara Rao Professor, for his guidance and support, K.L.UNIVERSITY, Vaddeswaram, Guntur . I also acknowledge the support and encouragement given by Dr.V.Srikanth Professor, Head of the Department of Computer Science and Engineering and my friend K.V.D.Krian and well-wishers to pursue the Research work.

## 7. REFERENCES

- [1] Hui Guan, Tingmei Wang, Weiru Chen "Exploring Architecture-Based Software Reliability Allocation Using a Dynamic Programming Algorithm" Inproc. 2<sup>nd</sup> symposium Int'l computer science and computational technology, Huangshan, P.R.China, 26-28, Dec .2009, pp.106-109.
- [2] D.L.Parnas. "The Influence of Software Structure on Reliability". In Proc.1975 Int'l Conf. Reliability software, Los Angeles, CA, April 1975. pp. 358-362.
- [3] M.L.Shooman. "Structural models for software reliability prediction". In Proc. 2nd Int'l Conf. Software Engineering, San Fransisco, CA, October 1976, pp. 268-280.
- [4] M.E. HELANDER, M. Zhao and N. Ohlsson. "Planning Models for Software Reliability and Cost". IEEE Trans. on Software Engineering, 1998, 24(6):420-434.
- [5] F. Zahedi and N. Ashrafi, "Software Reliability Allocation Based on Structure, Utility, Price and Cost". IEEE Trans on Software Engineering, 1991, 17 (4):345 - 356.
- [6] B. Boehm , R. Valerdi , J A. Lane et al, COCOMO Suite Methodology and Evolution, CrossTalk, 2005, pp. 20 - 25.
- [7] C. Y. Huang, J. H. Lo and S Y. Kuo, "Optimal Allocation of Testing resource Considering Cost, Reliability, and Testing Effort", In Prof. 2004 Pacific Rim Dependable Computing, French Polynesia, 2004, pp.103 - 112.
- [8] S. Y. Kuo, C. Y. Huang and M R. Lyu, "A Framework for Modeling Software Reliability , Using Various Testing Efforts and Fault Detection Rates". IEEE Transactions on Reliability, 2001, 50(3) :310 - 320.

- [9] A. Mettas, Reliability allocation and optimization for complex systems. In Proc. Annual Reliability and Maintainability Symposium, Los Angeles, CA, January 2000, pp.216-221.
- [10] R. W. Bulter and G.B. Finelli, “The infeasibility of quantifying the reliability of life-critical real-time software”, IEEE Trans. on Software Engineering, 1993,19:3-12.
- [11] M.R.Lyu. Handbook of Software Reliability Engineering. IEEE Computer Society Press, New York, 1996, pp.36.
- [12] M. R. Lyu. Handbook of Software Reliability Engineering. IEEE Computer Society Press, New York, 1996, pp.315