

**INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH
TECHNOLOGY****EVALUATION OF RELIABILITY FOR COMPONENT BASED SOFTWARE SYSTEM**

Parul Raheja*, Dr P.K Suri

ABSTRACT

In this paper we present a reliability simulation in component based applications. In component-Based Software Engineering we can treat a software system as a combination of interdependent components. The Component_Rel_Sim Algorithm is used to identify reliabilities of components. The simulation is done on the data generated by the Normal distribution, which describes the nature of failure process. Once the required data is generated the overall reliability of the system is calculated. The estimation of component reliability gives the overall reliability of the system.

KEYWORDS: Component Based Software, CBSE, Simulation, Reliability.

INTRODUCTION

Software application reliability is defined as the probability of a given system performing its task adequately for a specified period of time under the expected operating conditions. The probability that software will provide failure-free operation in a fixed environment for a fixed interval of time[6]. In every dimension of industry, education, research and development software reliability plays a critical part. Error free software goes under extensive debugging and testing to meet the user needs. This can be time consuming and costly. So component based software reliability development can be used. In component Based Software Development (CBSD) Commercial off- the-shelf (COTS) components are combined together to make a larger application. Reusability is one of the important characteristics of component Based Software Development. It offers higher quality, more reliable software. The main reason is that reusable components have been tested and therefore their quality can be assured. Thus if we are building software from existing components by assembling components will save lot of time and money. We used in this paper Component-Dependency Graph is directed graph in which vertices represent the components and edges represent transition from one state to another state. Based on Component-Dependency Graph we construct Component_Rel_Sim Algorithm

RELATED WORK

In[1], a component based software reliability model is introduced. This model estimates the reliability of component software with usage ratio of each component. The model transformed the execution path model into a component based model. In[2], approach based on failure dependencies between software components for assessing reliability of compound software is introduced. Accepted upper Bound for probabilities is calculated based on the principles of Bayesian hypothesis testing and include these into the reliability models. In[3], Discrete-event simulation is used as an alternative to analytical models like Markovian and semi-Markovian methods suffer from several limitations they cannot take into account the influence of various parameters such as reliability growth of the individual components, dependencies among the components, etc., in a single model. On the other hand Discrete-event simulation capture all the parameters due to its flexibility. In[4], Scenario-Based Reliability Estimation algorithm is discussed to analyze the reliability of component-based applications. This algorithm is applicable at the early development phase of a component-based software. It does not consider failure dependencies between components. In[5] the new approach is introduced for reliability of component based software systems. By determining all the independent execution paths of the software component reduces the number of terms in the overall reliability model which leads to the appreciable reduction in computation time and also it increases the probability of obtaining more accurate result. The upper and lower bound on the reliability is estimated of COTS component based software application systems. This is based on scenario analysis with the help of control flow graph and component dependency graph. In[6], rate-based simulation technique is used with analytical models to give effective methods for reliability measures. Functional dependency

and error correlation among components are two methods used for analyzing reliability of component-based software system.

SYMBOL USED

CDG: Component Dependency Graph

$\{N_j\}$: Set of j nodes in CDG

$\{E_i\}$: Set of i Edges in CDG

Interact (C_j, C_i): Number of times C_j interacts with C_i during a course of execution.

$TNC_{j,i}$: Total number of interactions among all components during a Course-of-execution

$NIC_{j,i}$: Probability of Transition from component i to component j.

E: Number of courses of execution (paths)

PE_k : Probability of execution of path E_k

N: Total Number of Components

$PRTR_{j \rightarrow i}$: Probability of Transition from Component j to Component i

MPOT: "Probabilities of Transitions" Matrix

$IPOT_{j \rightarrow i}$: Imperfect "POT" from j to i

MIPOT: Imperfect "PoT" Matrix

$RELC_j$: Reliability of Component j

VROC: "Reliabilities of Components" vector

STERM: Successful Termination

UTERM: Unsuccessful Termination

REL: Overall reliability of the application.

TSRUNS: Total number of simulation runs

PROPOSED WORK

Reliability using component based development is already in the literature. In the present work, reliability of component algorithm is proposed. The path of course of execution is based on random number $NDEXP$ generated using normal distribution. Normal distribution generate random numbers whose distribution is not known. Special case of normal distribution is known as standard normal distribution or the unit normal distribution if $\mu=0$ and $\sigma = 1$. In component dependent graph we don't know the execution path that is why we used normal distribution. In the graph $\langle N_j, E_i \rangle$ consist of j number of nodes and i number of vertices.[4]Number of interaction between two component given by

$$PRTR_{j \rightarrow i} = \sum_{k=1}^E PE_k * \left(\frac{|\text{Interact}(C_j, C_i)|}{|\text{Interact}(C_j, C_i)|_{i=1,2,\dots,N}} \right) \quad (1)$$

The sum of transition probabilities from any component should be unity. [8]Imperfect "Probabilities of Transition" matrix MIPOT is computed using values in MPOT and vector VROC (VROc is a vector that holds the reliabilities of all the components) as shown below

$$IPOT_{i \rightarrow j} = RELC_j * PRTR_{j \rightarrow i} \quad \text{for all } j, i. \quad (2)$$

Using the values in Imperfect State Transition matrix we generate another matrix called Cumulative State "Probabilities of Transition Matrix (MCPOT). when j=1 simulation starts from first component. Using normal distribution decide which component the next in the transition. From step 10 to 12 repeat until reached the terminal element. In step 11 calculate the overall reliability.

COMPONENT_REL_SIM ALGORITHM

Step 1: Find the Reliability of all Components($RELC_j$), Reliability of all transitions, Number of times C_j interacts with C_i (Interact (C_j, C_i)) and Total number of interactions among all components during a course of execution ($TNC_{j,i}$).

Step 2: Find Probability of Transition from component j to component i($PRTR_{j \rightarrow i}$).

Step 3: a) Using Reliabilities of all Components i.e.($RELC_j$) create Reliabilities of Components vector "VROC"

b) Using Probability of Transition from component j to component i create “Probabilities of Transitions” matrix MPOT.

Step 4: Now Initialize counters SC for successful termination of a particular Course-of-Execution & UC for unsuccessful termination of a Course-of-Execution.

Step 5: Read in VROC, MPOT, TSRUNS

Step 6: Using Equation number 2 calculate Imperfect “Probabilities of Transitions” matrix MIPOT

Step 7: Calculate Cumulative “Probabilities of Transitions” matrix MCPOT

Step 8: TSRUNS times Repeat steps 9 to 13

Step 9: j=1

Step 10: Now generate Random number from normal distribution. If $\mu = 0$ and $\sigma = 1$, the distribution is called the standard normal distribution denoted by $N(0,1)$ and a random variable with that distribution is a standard normal deviate.

Step 11: In this step, apply normal distribution

$$NDEXP = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$$

Step 12: Select a Course-of-Execution as follows:

If ($0 < NDEXP \leq MCPoT_{j \rightarrow 1}$)

j = 1;

Go to step 9.

Else

If ($MCPoT_{j \rightarrow 1} < NDEXP \leq MCPoT_{j \rightarrow 2}$)

j = 2;

Go to step 9.

.

.

.

.

.

.

Else

If ($MCPoT_{j \rightarrow N-1} < NDEXP \leq MCPoT_{j \rightarrow N}$)

j = N;

Go to step 9.

Else

If ($MCPoT_{j \rightarrow N} < NDEXP \leq MCPoT_{TERM}$)

SC = SC + 1 (Path terminates successfully).

ELSE

UC = UC + 1 (Path terminates unsuccessfully).

END IF

Step 13: Now calculate Reliability of any application

$$REL = \frac{SC}{TSRUNS}$$

Step 14: End

RESULTS

In this section, we have implemented the Component_Rel_Sim Algorithm for calculating the reliability of overall application using component based development. The overall reliability of the system to some extent depends on individual reliability. In this we assume that we know the reliability of each component, using this we calculate overall system reliability. In Component Dependency Graph (CDG) when we go from one component to another component during Course-of-Execution transition takes place but last component have not any transition. Normal Distribution is used in the algorithm

Using this algorithm we can investigate the reliability of system as a function of reliability of one component while others component remain constant.

Component	Com1	Com2	Com3	Com4
0.5	0.55	0.53	0.52	0.51
0.6	0.5679	0.5544	0.5533	0.5522
0.7	0.643	0.6321	0.63	0.6111
0.8	0.7111	0.7101	0.7098	0.7088
0.9	0.8001	0.7999	0.7654	0.7544
1	0.945	0.9444	0.9323	0.9213

Table 1: For different Components Overall System Reliability for various Reliabilities values

First column shows the reliabilities of each component. Each of the following column shows the reliability of the overall system while the reliabilities of all other component remain constant. For example the value 0.55 at the intersection of second column and second row of the table is the overall system reliability when reliability of the component C1 is 0.5 while reliabilities of all other components are kept as 1.0.

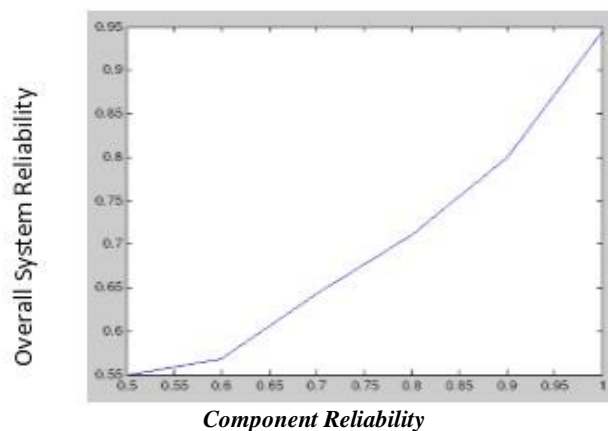


Figure 1: Reliability of Component C₁ v/s Overall Reliability

Figure 1 show the reliability of C₁ with respect to application reliability when the value of overall reliability X=(0.5000,0.6000,0.7000,0.8000,0.9000,1.0) and C₁= (0.5500,0.5679,0.6430,0.7111,0.8001,0.9450). Overall Reliability increases when the component reliability increases keeping all other reliability constant.

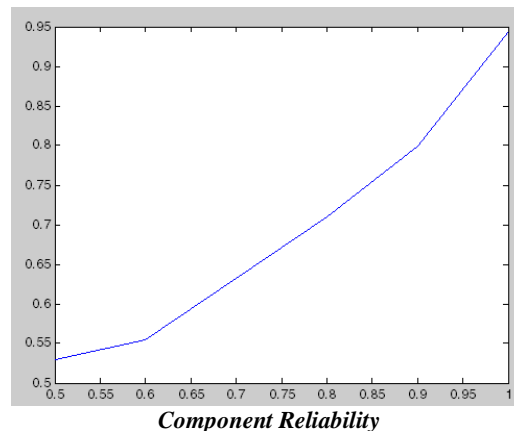


Figure 2: Reliability of Component C₁ v/s Overall Reliability

Figure 2 show the reliability of C_2 with respect to application reliability when the value of overall reliability $X=(0.5000,0.6000,0.7000,0.8000,0.9000,1.0)$ and $C_2 = (0.53,0.5544,0.6321,0.7101,0.7999,0.9444)$. When all other component reliabilities remain constant overall reliability increases when the component C_2 reliability increases.

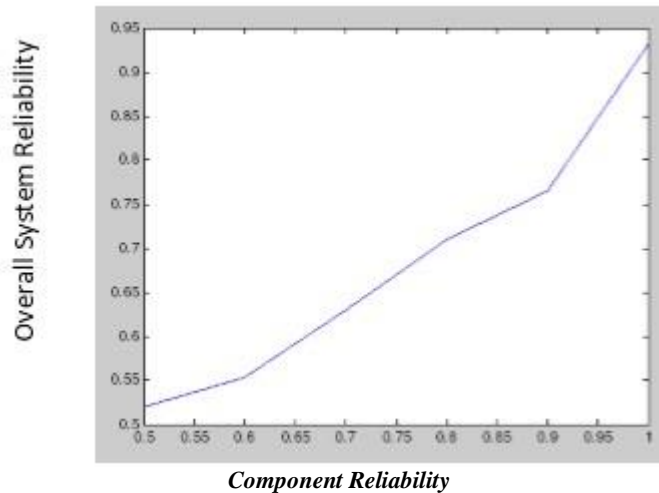


Figure 3: Reliability of Component C₃ v/s Overall Reliability

Figure 3 show the reliability of C_3 with respect to application reliability when the value of overall reliability $X=(0.5000,0.6000,0.7000,0.8000,0.9000,1.0)$ and $C_3=(0.52,0.5333,0.6300,0.7098,0.7654,0.9323)$. Overall Reliability increases when the component C_3 reliability increases.

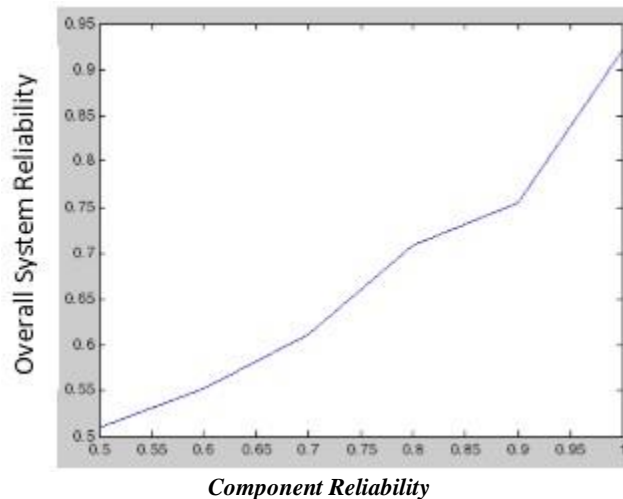


Figure 4: Reliability of Component C₄ v/s Overall Reliability

Figure 4 show the reliability of C_4 with respect to application reliability when the value of overall reliability $X=(0.5000,0.6000,0.7000,0.8000,0.9000,1.0)$ and $C_4=(0.5100,0.5522,0.6111,0.7088,0.7544,0.9213)$. Overall Reliability increases when the component C_4 reliability increases all other reliability remain constant.

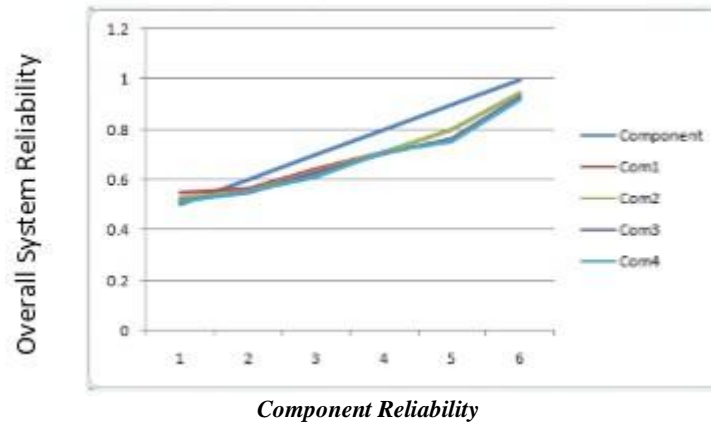


Figure 5: Component Reliability v/s Overall Reliability

Now Figure 5 shows the comparative analysis of different components. This can help when we are fitting the exciting component in the new application. It will tell which component is best fit. Each component reliability effect the whole new system. System reliability is calculated using how many simulation runs is done.

CONCLUSION

The simulator "Component_Rel_Sim" is designed and implemented successfully in this paper to estimate the reliability of the system using individual component reliability. The simulator designed here can help to decide whether the new component incorporate into new system based on the reliability. Also, there is effect of increasing number of components on the system. The number of components increase the overall reliability of system starts decreases. this is because in component dependency graph the number of transition path in the system increases.

REFERENCES

- [1] Dolbec, J. and Shepard, T. (1995) "A Component Based Software Reliability Model", Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research, Toronto, 7-9 November 1995, 19.
- [2] Monica Lind Kristiansena, Bent Natvigb, and Harald Holonec, "A Component-based Approach for Assessing Reliability of Compound Software" In .PSAM&ESREL (ed.), 11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference 2012, 25-29 June 2012, Helsinki, Finland . Curran Associates, Inc. ISBN 978-1-62276-436-5. KAPITTEL.s 1823 – 1832
- [3] Gokhale, Swapan S., Michael, R. Lyu and Trivedi, Kishore S., "Reliability Simulation of Component Based Software Systems," In the proceedings of 9th International Symposium on Software Reliability Engineering, 4-7th Nov. 1208, pp. 192-201.
- [4] Yacoub, S., Cakic, B. and Ammar, H., "Scenario- Based Reliability of Component Based Software," IEEE Transactions on Reliability, 53(4):465-480.
- [5] R. Chinnaiyan & S. Somasundaram, "Reliability Estimation Model for Software Components Using CEP" International Journal of Mechanical and Industrial Engineering (IJMIE), ISSN No. 2231 –6477, Volume-2, Issue-2, 2012.
- [6] ANSI/IEEE, 1991 Standard Glossary of Software Engineering Terminology, STD-729-1991.
- [7] Ruohao Huang , Michael R. Lyu," Simulation Techniques for Component-Based Software Reliability Modeling with Project Application"
- [8] P K Suri and Sandeep Kumar, "Design of Simulator for Reliability Estimation of Component Based Software System" IJCSNS International Journal of Computer Science and Network Security,9;9,(2009), pp.161-167.