A new path based reliability approach for estimation of reliability of Component Based Software Development

Lata Nautiyal*, Dr. Neena Gupta**, Dr. Sushil Chandra Dimri***

*Department of Computer Applications
Graphic Era University
Dehradun, India
lata.nautiyal1903@gmail.com

**Department of Computer Sciences
Kanya Gurukul Campus, GKV
Dehradun, India
neena71@hotmail.com

*Department of Computer Applications
Graphic Era University
Dehradun, India
dimri.sushil82@gmail.com

***Abstract*—One of the motivations for stipulating software architectures explicitly is the use of high intensity structural design information for improved control and prediction of software system quality attributes. Gradually software applications are mounting more complex and with more stress on reuse. The focus of this paper is to provide an outline for the state of the art of Component Based Systems reliability estimation and to present an approach for determining the reliability of component-based software architectures. The proposed model in this paper provides software system dependability in term of the reliability of different execution paths and the usage ratio of each component. The reliability of a component most powerfully depends on its environment. Therefore, we promote a reliability model parameterized by mandatory component reliability in a deployment perspective.**

Keywords-*Software reliability, Probability distribution, weighted mean, Component reliability, component-based software development.*

I. INTRODUCTION

Software reliability is one of the key metrics for determining the eminence of software. It is often defined as the probability of a failure-free operation of a computer program within a specified exposure time interval. Most of the analytical models, developed for measuring reliability, focus on observing the behavior of software, based on an operational profile and not on software architecture. Software architecture is defined as the structure of software at an abstract level, consists of a set of components, connectors and configurations. Modern software often embodies complex heterogeneous construction to achieve multiple quality requirements, such as the use of a parallel architecture to increase performance and/or introduce a back-up component to provide fault tolerance. In the new era of science where software are required almost all aspects of life, we can't imagine life without software. We have gigantic complex systems, massive data handling, record keeping everywhere software is required, Now the question comes how we can develop reliable software in less amount of time ?,the answer is the Component based software engineering . But the reliability of the software is a big question, the Software system failure occurs daily but in some cases the failures are not costly but some time the cost cannot be compensated, this makes reliability of the system very important.

II. COMPONENT RELIABILITY AND ITS CONTRIBUTION TO SYSTEM RELIABILTY

The reliability of a component can be uttered as a function from an input allotment or operational outline to a number between 0 and 1, as mentioned in [1]. In this paper, we presume that individual component reliability is known in advance. One of the methods to estimate component reliability is by identifying operational and logical errors and then predicting component reliability based on them [2]. This approach is useful when no documentation and source code is available, and can be used effectively when components are basically program functions. Another loom is based on service architecture of a component, measuring component reliability on the basis of the services used by that component where involvement of each reliability service is weighted by relative frequency of invocation of that service [3]. Overall system reliability is a function of individual component reliability; hence one of the critical tasks in estimation of overall system reliability is to determine precise individual component reliability [4].

A. *Reliability of Software*

Reliability of software is defined as "the likelihood of execution without failure for some definite interval of natural units or time" [5]. Reliability is an operational theory, and can be measured by execution. Reliability of COTS components based Software System can be resulting from reliabilities of individual COTS components which are there in the system [6]. Thus it can be concluded that the overall system reliability is a function of individual component reliabilities.

These days, Commercial Off the Shelf (COTS) components are used recurrently to develop the large software systems. Large scale use of COTS components has raised questions on the component's reliability, and the reliability of the system. This paper tries to calculate approximately the overall reliability of the COTS components based Software Systems. [7]

COTS components are "components which are bought from a third-party and incorporated into a system" [8] [9]. This reliability model should predict reliability of overall system based on individual reliability of COTS components integrated into a larger system. The approach proposed in this paper takes into the consideration the contribution of a component's reliability and path reliability to overall system reliability depending upon its usage time of components and on the executions paths, when the execution of overall system takes place. [10]

In this in paper we discuss how overall system reliability can be calculated from execution path reliability. The concept of usage time ratio is also discussed, also a mathematical model is proposed for calculation of reliability of overall system.

Mathematically, if R (C i) represents reliability of i^{th} component present in the system and R(S) represents overall system reliability then R(S) can be expressed as,

$$R(S) = \psi \{R(C_1), R(C_2), ---, R(C_n)\}$$

B. *Component Usage Ratio*

Component usage ratio is the ratio of a particular component execution time over the total software system execution time [11]. The value of the component usage ratio is $0 < U_{i,j} < 1$.This ratio can be calculated if the values of total component execution time (ti) and total software system execution time (Ts) are known for a component i. The component usage ratio weights the impact of component reliabilities on the overall software system reliability. As a general rule, the reliability of a component frequently executed is expected to have more impact on the overall system reliability than a component rarely executed [12][13].Therefore, the component which consumes most of the total execution time of the overall system is expected to have larger impact on the overall system reliability than a component with a very small execution time.

C. *Path Propagation Probability*

Once component usage ratio is calculated for all individual components, their involvement to overall system reliability cannot be directly inferred from this information [14]. This is because a system consisting of COTS components may have a number of possible paths of execution and these paths can have a probability associated with them i.e. probability that a particular path shall be executed. This factor further affects the contribution of individual component reliability to overall system reliability [15].

Path Propagation Probability is the probability that a particular path of execution will be taken, out of number of possible paths. This involves activation of certain components, based on path taken during an execution of overall system [16]. The path propagation probability depends on the decisions taken at component level which then decide the due course of that path. Hence path proliferation probability is inherently dependent on output probability of the decisions taken at component level [17].

III. PROPOSED METHOD FOR RELIABILITY ESTIMATION OF OVERALL SYSTEM

In this proposed model we used the idea of path propagation probabilities to estimate overall system reliability, which takes into consideration the contribution of the components that get activated during an execution rather than taking the contribution of inactive components.

In this paper an approach is proposed for calculating component usage ratio which does not vary with different sets of inputs. Hence the reliability predictions are more accurate and versatile

Path Propagation Probability (pi), this factor makes the reliability estimation more precise by taking into consideration different possible paths of execution.

The overall system reliability R(S) is estimated for a system with individual component reliabilities denoted by R (Ci) as follows.

Let the total time period of the execution of the software is T(S), in total N number of times the software process the input data in its lifetime is fi times the component i participated in the execution process if the average time taken by component i is ti for $T(C_i) = t_i \times f_i$ different set of inputs then the total usage time of component i is

And

$$T(S) = t_1 \times f_1 + t_2 \times f_2 + t_3 \times f_3 + ------ + t_n \times f_n \quad (1)$$

Where n is the total number of component in software system.

Then the usage ratio U_i of the component i is

$$U_i = T(C_i) / T(S) \quad (2)$$

Also

$$f_1 + f_2 + ----- + f_n = N \quad (3)$$

$$U_i = T(C_i) / T(S) = t_i \times f_i / (t_1 \times f_1 + t_2 \times f_2 + -- + t_n \times f_n) \quad (4)$$

Let the total numbers of available paths (Components in sequence) for all types of inputs to the software are K, identified with $P_1, P_2, ---- P_K$, it is totally random that which path will be the path of execution for the current instant input, this information is available in form of a probability distribution like given in the table

TABLE I. PATH PROPAGATION METHOD

Path of Execution	P_1	P_2	------------------	P_K
Probability of path selection	p_1	p_2	--------------------	p_k

Where

$$\sum_{i=1}^{K} p_i = 1$$

Let the number of component in path P_i are I_i, the number of component in paths P_1, P_2 ---, P_K can be given by $I_1, I_2, ---- I_K$.

Let the probability of failure of component C_j is $f_{i,j}$, where $C_j \in P_j$ then the probability of non failure is (1- $f_{i,j}$) and hence the probability that the path will survive $\prod_{j=1}^{I_j} \{(1 - f_{i,j})\}$.

Let the total N number of processes have been processed by the software in its life time and out of these n_1, n_2 -- n_k are the frequency of the path $P_1, P_2, --- P_k$ then of course

$$n_1 + n_2 + -- + n_k = N$$

Where $n_1 = p_1.N, \quad n_2 = p_2.N, \quad n_k = p_k.N$,

Since reliability of the system depends on the reliability of its component i.e.

$$R(S) = \psi \{R(C_1), R(C_2), ---, R(C_n)\}$$

and

$$R(P_i) = \zeta \{R(C_i), where C_i \in P_i\}$$

So

$$R(S) = \varphi \{R(P_1), R(P_2), ---, R(P_K)\}$$

Then with simple algebra reliability of the system can be given by $R(S)$ may be the weighted mean of $RP_1,\ RP_2, ---,\ RP_K$.

So = $R(S) = f_1 \times RP_1 + f_2 \times RP_2 + --- + f_k \times RP_K \ / \ (f_1 + f_2 + --- + f_K)$

or

$R(S) = p_1 \times RP_1 + p_2 \times RP_2 + ---- + p_k \times \mathrm{RP}_K$.

Where $p_i = f_i \ / \ N$

A. Numerical Simulation

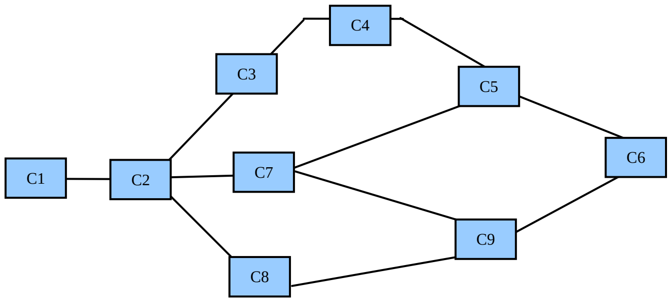

Fig. 1- Path Propagation Method

From figure 1 the existing paths are

$\mathrm{P}_1(\mathrm{c}_2 - \mathrm{c}_3 - \mathrm{c}_4 - \mathrm{c}_5 - \mathrm{c}_6),\ \mathrm{P}_2(\mathrm{c}_2 - \mathrm{c}_7 - \mathrm{c}_5 - \mathrm{c}_6),\ \mathrm{P}_3(\mathrm{c}_2 - \mathrm{c}_7 - \mathrm{c}_9 - \mathrm{c}_6)$ and $\mathrm{P}_4(\mathrm{c}_2 - \mathrm{c}_8 - \mathrm{c}_9 - \mathrm{c}_6)$,

Let the failure probability of all the components which are in the system are 0.02 then we have

$$R(P_i) = \prod_{j=1}^{Ij} \{(1 - f_{i,j})\}$$

$$R(P_i) = \prod_{j=1}^{Ij} (1 - 0.02)$$

$$R(P_i) = \prod_{j=1}^{Ij} (0.98)$$

$$= (0.98)^{Ij}$$

From figure

$$R(P_1) = (0.98)^5 = 0.9039$$

$$R(P_2) = (0.98)^4 = 0.9223,\ R(P_3) = (0.98)^4 = 0.9223,\ R(P_4) = (0.98)^4 = 0.9223$$

Thus the reliability of the system can be given by

Since, so

$$R(S) = f_1 \times R(P_1) + f_2 \times R(P_2) + ---- + f_k \times R(P_K) / (f_1 + f_2 + -- + f_k).$$

$$R(S) = f_1 \times R(P_1) + f_2 \times R(P_2) + f_3 R(P_3) + f_4 \times R(P_4) / (f_1 + f_2 + f_3 + f_k).$$

$$R(S) = p_1 \times R(P_1) + p_2 \times R(P_2) + p_3 \times R(P_3) + p_4 \times \mathrm{R}(\mathrm{P}_4).$$

$$R(S) = 0.2 \times (0.98)^5 + 0.3 \times (0.98)^4 + 0.1 \times (0.98)^4 + 0.4 \times (0.98)^4$$

$$R(S) = 0.18078 + 0.73784 = 0.91862 \textit{ i.e. the system reliability is } 91.862\%$$

4. Conclusions

We have presented an innovative approach to predicting the reliability of component-based software architectures. Also we have estimated 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. The proposed approach gains a sound foundation through its use of operational profile, putting it to a very different use. Moreover the rigidity in calculation of component usage ratio where it does not change with varying sets of inputs, lays emphasis on how this approach can be used for real time reliability prediction.

Acknowledgment

From the core of my heart, I am really thankful to the Management of Graphic Era University for always being supportive. I would also like to express my gratitude to the management of Gurukul Kangri University for providing such a commendable research oriented platform to us.

References

[1] J. Voas, J. Payne, R. Mills, J. McManus, Software testability: An experiment in measuring simulation reusability, In Proceedings of ACM Sigsoft, Seattle, WA., April 29-30, 1995.

[2] P. Brereton and D. Budgen, Component-based systems: A classification of issues, IEEE Computer, 33 (11), pp. 54-62, November 2000.

[3] L. Brownsword, T. Oberndorf and C. A. Sledge, Developing new processes for COTS-based systems, IEEE Software, 17(4), pp. 48-55, July 2000.

[4] Schneidewind, Norman F., Methods for Assessing COTS Reliability, Maintainability, and Availability, In Proceedings of the International Conference on Software Maintenance, Bethesda, MD, pp. 224-225, November 1998.

[5] Musa John, Software Reliability Engineering, McGraw-Hill, New York, NY, 1998.

[6] D. Hamlet, D. Mason, and D. Woit, Theory of software component reliability, In Proceedings of 23rd International Conference on Software Engineering, Toronto, Canada, May 2001.

[7] Tirthankar Gayen and R. B. Misra, Reliability Assessment of Elementary COTS software component, International Journal of Recent Trends in Engineering, 1(2), May 2009.

[8] John D. McGregor, Judith Stafford and Il-Hyung Cho, Measuring Component Reliability, In Proceedings of the 6th ICSE Workshop on Component-Based Software Engineering, May 2003.

[9] Shukla, R., Carrington D. and P. Strooper, Systematic operational profile development for software components, In Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), pp. 528-537,November-December 2004.

[10] George E.P. Box, Gwilym M. Jenkins and G.C. Reinsel, Time Series Analysis Forecasting and Control, San Francisco, California, U.S.A.: Holden-Day, 1976.

[11] R.C. Cheung, A User-Oriented Software Reliability Model, IEEE Transactions on Software Engineering, SE.-6(2) (1980), 118-125.

[12] A.L. Goel, K. Okumoto, A Time-Dependent Error Detection Rate Model for Software Reliability and Other Performance Measure, IEEE Trans. On Rel., 28 (1979), 206-211.

[13] S.S.Gokhale, K.S. Trivedi, Analytical Models for Architecture-Based Software Reliability Prediction: A Uni_cation Framework, IEEE Transactions on Reliability, 55(4) (2006), 578-590.

[14] S.S. Gokhale, Architecture-Based Software Reliability Analysis: Overview and Limitations, IEEE Transactions on Dependable and Secure Computing, 4(1) (2007), 32-40.

[15] Katerina Goseva-Popstojanova, A.P. Mathur, K.S. Trivedi, Comparison of Architecture-Based Software Reliability Models, Proc. 12 th International Symposium on Software Reliability Engineering, (2001), 22-31.

[16] Vivek Goswami and Y.B. Achraya, Method for Reliability Estimation of COTS Components based Software Systems, Proceedings of 20th International Symposium on Software Reliability Engineering, ISSRE, (2009).

[17] S. Krishnamurthy and A.P. Mathur, On the estimation of Reliability of a Software System Using Reliabilities of its Components, Proceedings The Eighth International Symposium on Software Reliability Engineering, (1997), 146-155.