Method for Reliability Estimation of COTS Components based Software Systems

Vivek Goswami[1], Y.B.Acharya[2]

[1]*Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India*

[1]vivek_goswami@daiict.ac.in

[2]*Physical Research Laboratory, Navarangpura Ahmedabad, Gujarat, India*

[2]yacharya@prl.res.in

Abstract- **In this paper a relatively simple and implementable approach is proposed for reliability calculation of software systems which are constructed from COTS components. This approach estimates overall software system reliability based on the individual component reliabilities which are integrated to form the system. Moreover those components having longer execution time contribute more towards overall system reliability. Here we have used operational profile of a component for the calculation of component usage ratio, which uses operational profile for a very different purpose from its traditional use in software reliability engineering.**

***Keywords*-** **Commercial-off-the-shelf (COTS), Software Reliability, Component Usage Ratio**

I. INTRODUCTION

Now-a-days, Commercial Off the Shelf (COTS) components are used frequently in large software systems. Large scale use of COTS components has raised questions on the component's reliability, and the reliability of systems derived out of these components. This paper tries to estimate the overall reliability of the COTS components based Software Systems, following a strict reliability model. Corporate downsizing and decreased government budgets, as well as the spiraling costs of building and maintaining large software systems, have made necessary the reuse of existing software components [1]. Moreover, this paradigm shift to COTS components appears inevitable, necessating drastic changes to current software development and business practices [2, 3]. Hence we need a model for estimating system reliability, to have a solid foundation and rigid system. In terms of software, the proportion of COTS software in a typical system is beginning to overtake the percentage of custom software. When a system fails, it may well be the COTS software that caused the system to fail, given the well-publicized defect rates for acquired software. Now the question arises why the model should be a strict one, i.e. with more accuracy, because when COTS components are put to use in larger system, they might be the weakest link in the chain so their reliability evaluation should be done at par with the custom components [4].

COTS components are "components which are bought from a third-party and integrated into a system" [5]. This reliability model should predict reliability of overall system based on individual reliability of COTS components as well as the interaction between them when integrated into a larger system.

The approach proposed in this paper takes into the consideration the contribution of a component's reliability to overall system reliability depending upon its usage time when the execution of overall system takes place. In the following text, in section II, we discuss how overall system reliability can be calculated from individual component reliability. The concept of usage time ratio is also discussed. In section III, we discuss the idea of path propagation probability with reasoning needed for understanding the proposed approach. In section IV, the mathematical model is proposed for calculation of reliability of overall system.

II. COMPONENT RELIABILITY AND ITS CONTRIBUTION TO SYSTEM RELIABILTY

Reliability of software is defined as "the probability of execution without failure for some specified interval of natural units or time" [6]. Reliability is an operational concept, and can be measured by execution. Reliability of COTS components based Software System can be derived from reliabilities of individual COTS components which are present in the system. Thus it can be concluded that the overall system reliability is a function of individual component reliabilities. Mathematically, if R_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,

$$R_s = f(R_1, R_2, ...R_i, ...R_n) \qquad (1)$$

In this paper we present an approach to calculate overall system reliability as a function of individual component reliability. To calculate individual component reliability, if a component can be subjected to exhaustive testing then one can easily predict its reliability, but this is not possible in most cases. In the following sub-sections we discuss component reliability, related works in the area of component reliability measurement and also propose a new approach towards calculation of usage ratio of a component when it is integrated into a larger system based on the operational profile of the component.

A. Component Reliability

The reliability of a component can be expressed as a function from an input distribution or operational profile to a number between 0 and 1, as mentioned in [7]. In this paper, we assume that individual component reliability is known in advance. One of the methods to calculate component reliability is by identifying operational and logical errors and then predicting component reliability based on them [8]. This approach is more useful when no documentation and source code is available, and can be used effectively when components are basically program functions. Another approach is based on service architecture of a component, measuring component reliability on the basis of the services used by that component where contribution of each service reliability to component reliability is weighted by relative frequency of invocation of that service [9]. 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 accurate individual component reliability. Here we propose a reliability model by calculating the accurate contribution of each component's reliability to overall system reliability, assuming individual component reliabilities are known in advance. Thus following section on Component Usage Ratio is paid more attention in our approach.

B. Proposed Approach for Component Usage Ratio based on Operational Profile

Component usage ratio is the ratio of a particular component execution time over the total software system execution time. 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* (t_i) and *total software system execution time* (T_s) 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 [10]. 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. Let $U_{i,j}$ be the component usage ratio of a component i for a path j out of the number of different paths of execution in the system then,

$$U_{i,j} = \frac{t_i}{T_s} \tag{2}$$

Thus for a given execution path "j" of a software system, the summation of the usage ratios of all the components which gets activated during this execution is 1. That is,

$$\sum_{i=1}^{n} U_{i,j} = 1 \tag{3}$$

Determining execution time (t_i) for a component is a difficult task as it will vary with different sets of input. This is because, based on the input values, different sets of operations shall be executed in a component. For a set of operations executed in a component execution, time might be very low and for some other it might be very high and thus component usage ratio calculated could be different for different sets of operations executed. To standardize the calculation of usage time ratio, we propose a method based on operational profile of the component to calculate its execution time, so that usage time ratio is a fixed quantity and does not change with varying inputs or varying sets of operations executed. Hence system reliability calculation can be performed in one shot, rather than calculating it several times for varying sets of inputs and operations. Our approach overcomes the problem of obtaining different usage time ratio for a particular component.

The operational profile (OP), a quantitative characterization of how the software will be used, is essential in any software reliability engineering application. In this paper, we are putting it to use for a different task, i.e. calculation of Component Usage Ratio, primarily component execution time. As COTS components are rich in functionality so an operational profile can be developed for a component from many of the approaches proposed in [12, 13]. Rakesh Shukla et.al, [12] devised an approach for systematic development of operational profile for a component. They proposed a method which uses both actual usage data and intended usage assumptions to derive a usage structure and usage distribution, where a usage distribution represents probabilities of operations. So, after developing an operational profile for a component, its execution time can be estimated using the following approach.

Let OP_i represents an operational profile for a component i, i.e. OP_i is a set of operations with their respective occurrence probabilities. We try to create subsets of OP_i. These subsets contain operations which have almost or exactly equal occurrence probabilities. Here,

$$OP_i = \{s_1, s_2, s_3 \ldots\ldots, s_n\} \tag{4}$$

where $s_1, s_2, \ldots\ldots, s_n$ are sets of operations with similar or same probabilities of occurrence as $P(s_1)$, $P(s_1)$, $P(s_2)$,….., $P(s_n)$ respectively. Now we further divide these subsets based on the execution time of each operation. In this division, we form sets of operation with nearly same or equal execution time. If s_i is a set, we form its subsets as $s_{i1}, s_{i2}, \ldots s_{ik}, \ldots, s_{jm}$ based on execution time with all operations in subsets $s_{i1}, s_{i2}, \ldots s_{ik}, \ldots, s_{jm}$ having execution time $t_{i1}, t_{i2}, \ldots t_{ik} \ldots\ldots t_{im}$ respectively. Then execution time of the particular component can be expressed as,

$$t_{component} = \sum_{i=1}^{n} P(s_i) \sum_{j=1}^{m} \frac{|s_{ij}|}{|s_i|} t_{ij} \tag{5}$$

Here we consider the contribution of each set of operations execution time on the basis of their cardinality ratio with the superset and then the contribution of each element of operational profile is weighted by its occurrence probability $P(s_i)$. In this way, the total execution time for a component can be calculated and hence a standard Component Usage Ratio can be calculated.

III. PATH PROPAGATION PROBABILITY

Once component usage ratio is calculated for all individual components, their contribution to overall system reliability cannot be directly inferred from this information. 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. This is made clearer by the following definition of path propagation probability and its impact on reliability calculation of system.

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. The path propagation probability depends on the decisions taken at component level which then decide the due course of that path. Hence path propagation probability is inherently dependent on output probability of the decisions taken at component level. Let us take a simple example to illustrate path propagation probability. Assume a software system having six components (Fig.1). Component 2 is performing some mathematical calculation and its output is a numerical value which decides the path (or series of components) to be executed, as follows.

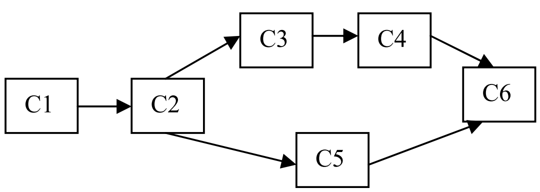

Fig. 1. Example Software System

If output of C2 is greater than or equal to zero, then path which shall be executed is C1→C2→C3→C4→C6 otherwise second possible path is C1→C2→C5→C6, and this probability that output of component 2 is greater than zero can be calculated easily based on input domain of mathematical functions forming component 2. Sometimes in a software system few paths are rarely executed, thus contribution of components present in a path which rarely gets executed should be lesser than those present in the path which is executed often. Therefore, component reliabilities of components which are present in particular path should be weighted by path propagation probability of that path. In this paper, we estimate reliability of a system based on possible paths of execution of the system. That is, if three paths are possible for execution of overall system and their Path Propagation Probabilities are p_1, p_2 and p_3 respectively then,

R_s = p_1(contribution of component reliabilities of path1) + p_2(contribution of component reliabilities of path 2) + p_3(contribution of component reliabilities of path 3)

Also,

$$\sum_{j=1}^{n} p_j = 1 \qquad (6)$$

After all possible paths of execution with their respective path propagation probability are determined, their sum should add up to 1 reflecting the complete coverage of all possible paths.

IV. PROPOSED METHOD FOR RELIABILITY ESTIMATION OF OVERALL SYSTEM

Reliability of a COTS components based software system estimated using the reliability of the components is an approach which is the basis of research in this area. Although the contribution of each individual component's reliability to the overall system weighted by its usage time was proposed in [10], the estimate in [10] gave the same importance to contribution of those components which are executed rarely as well as components which are executed many times. Hence our estimate is better than [10], as 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. Reliability Bounds were predicted using the Component Usage Ratio in [11], where maximum usage time was used and so diverted from estimation of actual reliability value. In this paper we proposed an approach for calculating component usage ratio which does not vary with different sets of inputs. Hence the reliability predictions are more accurate and versatile. We propose an approach which takes into consideration many important factors for reliability prediction such as

1. Component Usage Ratio (U_i), which weighs the contribution of individual component reliability to overall system reliability.
2. Path Propagation Probability (p_i), 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_i as follows.

$$R_s = \sum_{j=1}^{n} p_j \left(\sum_{i=1}^{m} U_{i,j} R_i\right) \qquad (7)$$

Overall system reliability is a summation of terms weighted by path propagation probability where each term is a summation of individual component reliability weighted by their respective Component Usage Ratio for that particular assembly of components of a given execution path.

This approach lends itself to implementation for calculating reliability of COTS components based Software System when reliability of individual components and their usage structures are known in advance.

V. CONCLUSION

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.

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] M. Vigder and J. Dean, An architectural approach to building systems from COTS software components, In *Proceedings of the 1997 Conference of the Centre For Advanced Studies on Collaborative Research*, Toronto, Ontario, Canada, November 1997.

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

[7] 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.

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

[9] 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.

[10] Jean Dolbec and Terry Shepard, A Component Based Software Reliability Model, In *Proceedings of the 1995 Conference of the Centre For Advanced Studies on Collaborative Research* ,Toronto, Ontario, Canada, November 1995.

[11] Tirthankar Gayen and R. B. Misra, Reliability Bounds Prediction of COTS Component Based Software Application, *International Journal of Computer Science and Network Security*, 8(12), pp. 219-228, December 2008.

[12] 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.

[13] John D. Musa, Operational Profiles in Software-Reliability Engineering, *IEEE Software*, 10(2), pp. 14-32, March 1993.