Reliability Modeling for Embedded System Environment compared to available Software Reliability Growth Models

Dr. Sapna Juneja
CSE Department
BMIET
Sonepat, India
sapnajuneja1983@gmail.com

Dr. Abhinav Juneja
CSE Department
BMIET
Sonepat, India
abhinavjuneja@yahoo.com

Rohit Anand
ECE Department
G.B.Pant Engg.College
Delhi, India
roh_anand@rediffmail.com

***Abstract*—Software plays a vital role in almost all of the embedded systems and systems that are mission-critical including aircrafts, automobile, nuclear power plants and various robotic medical applications [1]. Apart from the functional complexity of software, the associated reliability of embedded software systems has become an important area of concern. Various techniques are being devised to predict the quality of a software system from its designed structure, which include functional fields like overall reliability of the system, safety for human beings in life critical real time applications, operative environment and working performance. However, the findings of these quantitative structural measures depend on estimating the designing time for a group of predefined parameters. To sort out this problem, this paper presents a new design-time structural and optimization technique that includes uncertainties of parameters. The work mainly concentrates upon various structured reliability prediction models and the Embedded System. It proposes a model of reliability for predicting, calculating and controlling reliability of Embedded Software system where a series of parameters have to be predicted corresponding to various different uncertain factors.**

Keywords—Optimization, Reliability, SRGM

I. Introduction

As more and more software are becoming the part of Embedded Systems, we should verify that they do not insert errors into these systems. If not taken into consideration properly, reliability of the part of the software may harm the reliability of the complete system. Predicting actual reliability of software is not an easy job. More difficult the problem is, promising advances are still being made toward more reliable software development for the Embedded Systems. More high-level modules and superior processes are coined in the field of software engineering. By now most of the researchers and their developed technologies have been focussed on enhancing reliability of software or a hardware component in any environment and have succeeded to a large extent in achieving the targets set. But as technology is taking new heights and embedded systems are affecting our lives much more than previous times through new applications every day. So there is a need to think of novel methods to make embedded systems more reliable and robust. It is proposed to develop an efficient reliability model for estimating and measuring reliability of embedded systems in comparisons with the existing software reliability growth models using modern optimization techniques.

II. Available Models for Predicting the Reliability of Software

A. Jelinski Moranda De Eutrophication Model

It was the first available software reliability growth model [2] developed in 1972 and its main purpose was to predict the reliability of any software system. There are some assumptions on which this model is dependent, and these underlying assumptions are as:

a) At the start of the testing process there are some faults present in the system and these number of faults are fixed. It can be represented by μ_0.

b) Every fault is equally hazardous to the system in terms of Hazard Rate.

c) Faults in the system and time between failures are independent to each other.

d) Whenever any failure arises in the system, then the fault that is responsible for causing this failure has been immediately removed from the system and ensuring that it will not lead to insertion of some new faults into the system. Based upon these underlying assumptions, the hazard rate of the program upon the removal of the $(i-1)^{th}$ fault is directly proportional to

$$Z\ (\Delta t \mid t_{i-1}) = \Phi\ [\mu_0 - (i-1)] \qquad (1)$$

Here Z=Hazard rate of the system
Δt= Total time of testing the software
Φ= Hazard Rate of one fault within the software
μ_0 = Number of faults present in the system at the start of testing procedure
t_{i-1} = Time when i-1 fault has been removed from the software

B. Goel Okumoto Model

This model was proposed by Goel and Okumoto[3] in year 1979.The assumptions on which this model is based are:

a) The total number of failures in the system follows poisson distribution.

b) The amount of software failures is directly proportional to the assumed number of unexplored faults.
c) The number of failures in different time slots are not dependent.
d) When a failure has been located in the system, the bug that caused this failure has removed immediately without inserting any new bug into the software.

Now the Hazard rate of this model can be represented as:

$$Z(t) = a(1 - e^{-bt}) \quad (2)$$

Where [a >0, b>0]

a =Expected No. of Faults present in the software at the start of testing process.

b = Fault Detection Rate is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(1)", not "Eq. (1)" or "equation (1)", except at the beginning of a sentence: "Equation (1) is . . ."

C. NHPP Model

The assumptions on which it works are:

a) The faults present in the system at the start of testing process are equally distributed in the system in major faulty areas.
b) When a failure has been located in the system, the bug that caused this failure has removed immediately without inserting any new bug into the software.

The Hazard Function (Failure Rate) of this model [4] can be represented as:

$$Z(t)=[a-\Delta(t)]\ \varphi(t) \quad (3)$$

a = Expected number of faults present in the system at the start of testing process.
$\varphi(t)$ = Hazard Rate of one fault.

D. Littlewood Verral Model

It was given by Littlewood and verral[5] in year 1973.

He stated that software reliability is not dependent upon the no. of errors in the program but obtained by the times between the failures. This model also takes in to consideration the quality of the programmer and difficulty of the programming task. The Hazard Function (Failure Rate) of Littlewood Verral Model can be represented as:

$$Z(t)= (\alpha-1)(N^2 + 2B\Phi(\alpha-1))^{-1/2} \quad (4)$$

α=Programmer Quality and Task difficulty
N= No. of iterations of testing
B= Fault Reduction Rate
Φ= Failure Rate of one fault

E. Generalized Goel Model

It was proposed by A.K.Goel. It is also based on the concept that whenever the software is under test, the failure rate of the software does not remain constant, it initially increases and then decreases. This model handles varying failure rate ie first increasing and then decreasing failure rate of the system. The Hazard Rate [6] of this model can be represented as:

$$Z(t)= abe^{-btc}\ t^{(c-1)} \quad (5)$$

a = Expected no. of faults
b = Fault detection rate
c = Constant that represents the quality of testing

III. Proposed Model of Embedded System Reliability

The proposed model for representing software failures is Basic Random model. The purpose of introducing Basic Random Model is that till now so many reliability models has been developed that are used to predict the reliability of any software system, but no successful reliability model has been generated that computes or predicts the reliability of Embedded Systems. As now a days Embedded Systems are of more usage than standalone Software System but still the reliability of these Embedded Systems has been predicted by applying SRGM's which sometimes don't provide more accurate results [9]. So, in order to increase the accuracy for computing the reliability of Embedded System Basic Random model has been defined and implemented that takes in to consideration both hardware and software failures for predicting the reliability and provide more accurate results. So, in order to predict the reliability of Embedded System, the parameters of embedded system [7] has been imposed upon the Basic random Model [8]to predict the software reliability of the Embedded Systems. Then the the failure rates have been computed in two cases as:

1) First, apply the parameters of Embedded system on any of the available Software Reliability Growth Model and then compute the failure rate of Embedded System when its reliability had been predicted using growth model that has been designed to compute the reliability of Software only.

2) After that, apply the same parameters of Embedded System on Basic Random Model that has been designed to compute the reliability of Embedded System (Not for standalone software system or for standalone hardware system) and thcn compute the Failure Rate Again for the same Embedded System.

For our proposed model to be more accurate and precise, the rate of failure of Embedded System using proposed model (Basic Random Model) should be less than the rate of failure of same Embedded System using other Software Reliability Growth Models. So, in order to develop a new model for predicting reliability of embedded systems, we need to calculate the occurrence rate of software failures of the Embedded System software at any given time interval t. So, it can be calculating by adding the occurrence rate of pure software failures and occurrence rate of hardware related software failures. The basic random model is based on the hypothesis that software failures that occur due to failure hardware comes randomly without any pattern with a constant software or hardware. So we can specify whether the failure occurred is due to faulty hardware or due to failure in the software. Hence, we consider that the software failure occurred due to faulty hardware follows the uniform distribution. The failure rate in case of pure software failures can be identified by using the G-O model. Therefore, the failure rate of embedded system failures at any given time t / ES (t) is represented as the sum of the failure rate of hardware related software failures and pure software failures as:

$$\lambda_{ES}(t) = \lambda_{sw|hw} + \lambda_{sw}(t) = \lambda_{sw|hw} + abe^{-bt} \tag{6}$$

Where $\lambda_{sw|hw}$ is the constant occurrence rate of software failures that occurs due to hardware.

λ_{sw} is the occurrence rate of complete software failures.

a, b are the parameters taken from G-O mEan value function of the Embedded software system and can be represented as:

$$m_{ES}(t) = \int_0^t \lambda_{sw|hw} + \lambda_{sw}(t)dt = \lambda_{sw|hw} \times t + a(1 - e^{-bt}) \tag{7}$$

The above value can be obtained by applying the integration function on both types of failure rates ie failure rate of software caused by hardware as well the complete software failures. In the random based model, the reliability of the embedded software reliability for the time (t) to (t+x) where t≥0 and (t+x) ≥ 0 can be represented as:

$$R_{ES}(x \mid t) = e^{m_{ES}(t) - m_{ES}(t+x)}$$
$$R_{ES}(x \mid t) = e^{-(\lambda_{sw|hw} \times x + ae^{-bt}(1-e^{-bx}))} \tag{8}$$

Here $R_{ES}(x|t)$ is the Embedded system Reliability.

TABLE I. PARAMETER VALUES OF EMBEDDED SYSTEM TAKEN FOR THE EVALUATION OF THE MODELS

S.No.	*Parameter*	*Values*
1	Crashing Penalty Cost, C_p	2000
2	Amount of removal of a bug after its launch, C_R	200
3	Amount of removal of a bug before its launch, C_r	15
4	Cost of Testing during period [0, t], f(t)	20t
5	Number of iterations, N	200
6	Probability density function g(t) of the bug occurrence time follows an exponential distribution with parameter μ	1
The Random parameter X is considered to be uniformly distributed between 0 (U min) and 200 (U max)		

TABLE II. FAILURE RATES OF VARIOUS MODELS

Model	C_p	C_R	C_r	*COT*	*N*	*PDF*	*Failure Rate*
Goel Okumoto Model	2000	200	15	20t	200	1	631.3075
Jelinski Moranda Model	2000	200	15	20t	200	1	497
Littlewood Verral Model	2000	200	15	20t	200	1	223.4412
Generalized Goel Model	2000	200	15	20t	200	1	210.4872
NHPP Model	2000	200	15	20t	200	1	420.9744
Basic Random Model	2000	200	15	20t	200	1	126.2615

IV. CONCLUSION

Basic Random model is the underlying model which is closer to the embedded system operational environment prediction. In the paper, standard parameters of Embedded System has been taken and applied on all the six reliability models The parameters used for reliability estimation to determine the failure rate of different reliability models under same standard conditions. From the results of modeling using MATLAB, it can be easily concluded that the failure rate of basic Random Model is the lowest of all the models which were taken into consideration for the purpose of reliability evaluation. As the failure rate of Random Based Model is the least, it further implies that this Reliability Model is most suited for an embedded system environment as being most reliable of all the models employed under standard input conditions. Utility of this work is fruitful in the prediction of best reliability model for embedded system and according to my results the Basic Random Model for estimation of reliability of Embedded System is the most suitable model. If this model can be used before the final development and deployment of the software, it can lead to more accurate prediction results and can save the embedded systems from many kinds of failures, also it can avoid many unnatural disasters that are caused by the failure of the embedded systems as embedded systems are now a days being used in life critical systems..

REFERENCES

[1] Michael R Lyu:Software Reliability Engineering: A Roadmap.Future of Software Engineering,IEEE 2007.

[2] Ganesh Pai: A Survey of Software Reliability Models.University of Virginia 2002.

[3] ZuzanaKrajcuskova:SoftwareReliability Models.IEEE 2007.

[4] Shingeru Yamada, Hiroshi Ohtera,Hiroyuki Narihisa:IEEE Transactions on Reliability, Vol. R 35, No.1,1986.

[5] B. Littlewood , J. L. Verrall :A Bayesian Reliability Growth Model for Computer Software, Applied Stutistics. pp.332-346, 1973.

[6] A.L.Goel: Software Reliability Models- Assumptions, Limitations and Applicability, IEEE Transactions of Software Reliability 1985.

[7] Tanvir Khan:Optimization for software release and crash, B.S., Louisiana State University – Baton Rouge 2007

[8] Jinhee Park, Hyeon-Jeong Kim, Ju-Hwan Shin, Jongmoon Baik : An Embedded Software Reliability Model with Consideration of Hardware related Software Failures, IEEE Sixth International Conference on Software Security and Reliability,pp.207-14, 2012.

[9] Xin He,Yun Jun Li: Software Reliability Analysis on Embedded System Based on SFMEA and SFTA Model, International Conference on Systems and Informatics,pp. 2471-74,IEEE 2012.

[10] Jie Yu, Xuehai Zhou:A New Reliability Evaluation Method for Embedded Sysytem, IEEE, pp.224-30, 2008.

[11] Kelvin J Cronin: A Change Prediction Model for Embedded Software Applications, IEEE, pp.5-8,1998.

[12] Anne Immonen,Antti Niskanen: A Tool for Reliability and availability Prediction, Proceedings of the 2005 31st EUROMICRO Conference on Software Engineering and Advanced Applications, IEEE 2005.

[13] Paolo Giusto, Grant Martin :Reliability Estimation of Execution Time of Embedded Software, IEEE, pp.580-87, 2001.

[14] Nauremon Wattanapongsakorn: Reliability Optimization Models for EmbeddedSystems With Multiple Applications, IEEE TRANSACTIONS ON RELIABILITY, VOL. 53, No. 3,2004.