Reliability Evaluation of Web based Software

P. K. Suri[1], Bharat Bhushan[2]

[1]*Department of Computer Science & Applications, Kurukshetra University, Kurukshetra (Haryana) India,*
[2]*Department of Computer Science & Applications, Guru Nanak Khalsa College, Yamuna Nagar (Haryana), India*

Summary

The IEEE defines reliability as "the ability of a system or component to perform its required function under stated conditions for a specified process time". To most project & software development managers, reliability is equated to correctness, that is, they look to testing and the number of bugs found & fixed. Web based software consists of finite number of modules called web pages. The control is transferred among the web pages through hyper links according to Markov process [1, 2]. The software fails or reaches its final state successfully and/or moves to another state and finally enters into the terminal state. The reliability of web based software may be defined as the probability of entering of web based software into a terminal state successfully. The present paper attempts to analyze the reliability behavior of web based software system using Markov process. The software simulation technique designed here will act as a tool for the software quality assurance (SQA) team for evaluating the reliability of web based systems. The reliability evaluation acts as indicator of quantum the SQA needed for meeting the project software quality objectives.

Keywords: *Web based software, Simulation, Reliability, Markov process, web pages, Reliability models*

Introduction

Web based systems are playing an important role in modern computer savvy society today. Because of the pervasive nature and the massive user population, various existing software engineering approaches need to be adopted for web engineering. Web quality assurance (WebQA) is the application area which deals with analysis, testing, quality/reliability improvement for web-based applications.

Davila-Nicanor, L.; Mejia-Alvarez, P.[3] focused on the development of a methodology for the evaluation and analysis of the reliability of web-based software applications. They tested the methodology in a web-based software system and used statistical modeling theory for the analysis and evaluation of the reliability. The behavior of the system under ideal conditions was evaluated and compared against the operation of the system executing under real conditions. The evaluation and improvement process is performed in their methodology to evaluate and improve the quality of the software system.

J. Tian, S. Rudraraju and Z. Li [4, 6] discussed web usage and problems for web applications, evaluate their reliability and examine the potential for reliability improvement. Based on the characteristics of web applications and the overall web environment, they classify web problems and focus on the subset of source content problems. Using information about web accesses, they derive various measurements that can characterize web site workload at different levels of granularity and from different perspectives. These workload measurements, together with failure information extracted from recorded errors, are used to evaluate the operational reliability for source contents at a given web site and the potential for reliability improvement.

.J. Tian and L. Ma [5, 8] characterized the problems for web applications, examined existing testing techniques that are potentially applicable to the web environment, and introduce a strategy for web testing aimed at improving web software reliability by reducing web problems closely identified with web source contents and navigations whole by analyzing the dynamic web contents and other information sources not covered in our current case studies.

C. Kallepalli and J. Tian [6, 7] used statistical testing and reliability analysis effectively to assure quality for web applications. To support this strategy, they extract web usage and failure information from existing web logs. The usage information is used to build models for statistical web testing. The related failure information is used to measure the reliability of web applications and the potential effectiveness of statistical web testing.

L. Ma and J. Tian [9, 10] discussed defect classification and analysis framework, orthogonal defect classification (ODC), to analyze web errors and identify problematic areas for focused reliability improvement. Based on information extracted from existing web server logs, web errors are classified according to their response code, file type, referrer type, agent type, and observation time.

Web based software is a collection of web pages associated with hyperlinks to communicate with other components of the software. Each web page is considered

Manuscript received September 5, 2007
Manuscript revised September 20, 2007

as a functionally independent component of web based software. The evaluation is carried out using Markov analysis which looks at a sequence of events and analyses the tendency of one event to be followed by another. Using this analysis one can generate a new sequence of random but related events which look similar to the original. This Markov process is stochastic in nature which has the property that the probability of transition from a given state to any future state depends only on the present state and not on the manner in which it was reached [11].

If $t_0 < t_1 < t_2 < \ldots < t_n$ represents the points in time scale then the family of random variables $\{X(t_n)\}$ is said to be a Markov process provided it holds the Markovian property :

$$P\{X(t_n) = x_n \mid X(t_{n-1}) = x_{n-1}, X(t_0) = x_0\} = P\{X(t_n) = x_n \mid X(t_{n-1}) = x_{n-1}\} \quad \forall \quad X(t_0), X(t_1), \ldots, X(t_n)$$

Markov process is a sequence of n experiments in which each experiments has n possible outcomes $x_1, x2, \ldots, x_n$. Each individual outcome is called a state and probability (that a particular outcome occurs) depends only on the probability of the outcome of the preceding experiment. The simplest of the Markov processes is discrete and constant over time. It is used when the sequence of experiment is completely described in terms of its states (possible outcomes). There is a finite set of states numbered 1, 2, 3, ….n and this process can be only in one state at a prescribed time. The system is said to be discrete in time if it is examined at regular intervals eg. daily, weekly, monthly or yearly.

Transition Probability

The probability of moving from one state to another or remaining in the same state during a single time period is called transition probability .
Mathematically, the probability

$$P_{x_{n-1}, x_n} = P\{X(t_n) = x_n \mid X(t_{n-1}) = x_{n-1}\}$$

is called the transition probability. This represents the conditional probability of the system which is now in state x_n at time t_n provided that it was previously in state x_{n-1} at time t_{n-1}. This probability is known as transition probability because it describes the system during the time interval (t_{n-1}, t_n). Since each time a new result or outcome occurs, the process is said to have stepped or incremented one step. Each step represents a time period or any other condition which would result in another possible outcome. The symbol n is used to indicate the number of steps or increments.

The transition probability can be arranged in a square matrix form denoted by

$$P = \begin{pmatrix} p_{11} & p_{12} \ldots\ldots & p_{1n} \\ p_{21} & p_{22} \ldots\ldots & p_{2n} \\ \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} \ldots\ldots & p_{nn} \end{pmatrix}$$

Such that $\sum p_{ij} = 1$; i=1, 2, 3…..n and $0 \le p_{ij} \le 1$

Proposed model

Software reliability can be computed analytically with the help software reliability models. These models are based on some assumption for the simplification of the solution [12, 13, 14]. Modeling approaches discussed in [14, 15, 16] are extended and applied to web based software system. As we approach to more & more realistic and complex situations it becomes almost impossible to obtain an analytic solution. Then simulation techniques are used.

Assumptions

- The control flows from one web page to another web page according to Markovian property.

- The process of transition continues till it reaches a state called the terminal state. This ensures the successful completion of the transitions. TSTATE denotes the terminal state and 1, 2, 3 ….. n denotes transition states corresponding to web pages numbered as 1, 2, 3….n . Thus web software has state space {1, 2, 3….n, TSTATE}. The first page of the web software is called a home page and corresponds to initial sate 1.

- In the absence of a successful transition from page 1 to TSTATE successively, the process is probable to fail abruptly. Such a state, if it exists, is denoted by FSTATE.
- Each of transient web pages 1, 2, 3….n is prone to failure. The web page i have an associated reliability rel_i which is the probability that the web will operate correctly when invoked and will transfer the control to

other web page successfully as and when intended by the user. Therefore the probability of failure to enter from state i to j will be (1- rel_i). The Markov chain for the imperfect web software becomes (1, 2, 3,….n, FSTATE/ TSTATE). Therefore

$itpm_{ij} = rel_i * tpm_{ij}$ ∀ $i=1.2,-----n$ and ∀ $j=1, 2-----n$, TSTATE;

$itpm_{ifstate}, i = 1 - rel_i$ ∀ $j = 1, 2------n$;

$itpm_{tstate\ tstate} = 1$

$itpm_{ifstate, fstate} = 1$

Note that the two states viz. FSTATE and TSTATE are mutually exclusive and that any of the transient states *1,2, ----n* will eventually lead either to a FSTATE or a TSTATE .

System reliability is defined to be the probability that system eventually completes its task successfully without failing from transition from one page to the other till its termination state is reached.. When applied to our model it is simply the probability that the Markov chain for web software is eventually absorbed into TSTATE rather than the FSTATE.

Thus RELWEB = NAB / SIMRUN where NAB is the number of times the control goes to observing state out of SIMRUN simulation runs.

Terms and Notations

SIMRUN	:	number of simulation runs
REL	:	reliability vector
NT	:	counter for successful termination
Pij	:	probability that control will be passed next onto web page j from i
NPAGES	:	no of web pages in the website
TSTATE	:	terminal state
FSTATE	:	failure state
SYSREL	:	system reliability
TPM	:	ideal state transition probability matrix
TPMI	:	imperfect state transition probability matrix
CTPM	:	Cumulative state transition probability matrix
RELWEB	:	Reliability of web based software

Algorithm: WEB_REL_SIMULATOR

Step 1. [Initialize counter for successful termination.]
NT=0
Step 2. [Input a) Ideal state transition matrix (TPM) for web based software system
b) Reliability vector (REL)
c) Number of simulation runs (SIMRUN).]
Step 3. [Compute Imperfect State Transition Probability Matrix (TPMI =REL * TPM)]
Step 4. [Compute Cumulative state transition probability matrix (CTPM)]
Step 5. [Repeat SIMRUN step 6 through step 8]
Step 6. [Initialize the local current state counter (i=1)]
Step 7. [Generate a uniformly distributed random Number (say RANDOM.)]
Step 8. [Select the transition path using RANDOM and CTPM row corresponding to i.]

IF 0<RANDOM<= $CTPMi_1$
i=1;
go to step 7;
IF CTPM i1<RANDOM<=CTPMi2
i=2;
go to step 7;
.
.
.
.
.
.
.
IF CTPM i (n-1) <RANDOM<=TCPMin
i=n;
go to step 7;
IF CTPMin < RANDOM < = $TCPM_{Istate}$
NT=NT+1
ENDIF
Step 9. [Compute estimated reliability.]
RELWEB = NT/SIMRUN

Results

1) Inputs
Number of pages
Ideal transition probability matrix TPM of order n*(n+1)
Reliability vector REL
Number of simulation runs SIMRUN

Details of inputs (Reliability vector values) and output (Web based software Reliability, S.Rel)

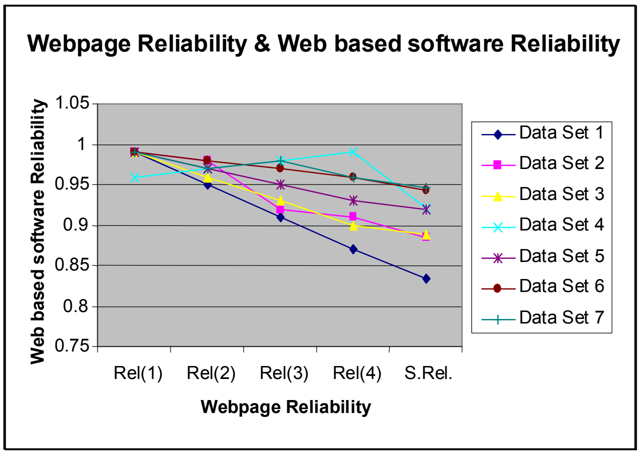
Page Wise Rel	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Data Set 6	Data Set 7
Rel(1)	0.99	0.99	0.99	0.96	0.99	0.99	0.99
Rel(2)	0.95	0.98	0.96	0.97	0.97	0.98	0.97
Rel(3)	0.91	0.92	0.93	0.98	0.95	0.97	0.98
Rel(4)	0.87	0.91	0.9	0.99	0.93	0.96	0.96
S.Rel.	0.84	0.88	0.89	0.92	0.92	0.94	0.95

Table (1)

Description of outputs

The above table gives a view of the system reliability computed for different set of values of webpage reliability by executing the simulator 100000 times for each set of inputs. It is found that the system reliability increases as the web page reliability increases
.

The following graph shows the trend related to webpage reliability and website reliability using the data mentioned in the table (1).



Graph (1)

2) Inputs

Number of pages
Ideal transition probability matrix TPM of order n*(n+1)
Reliability vector REL
Number of simulation runs SIMRUN

Details of inputs (Number of simulation runs) and output (Web based software Reliability)

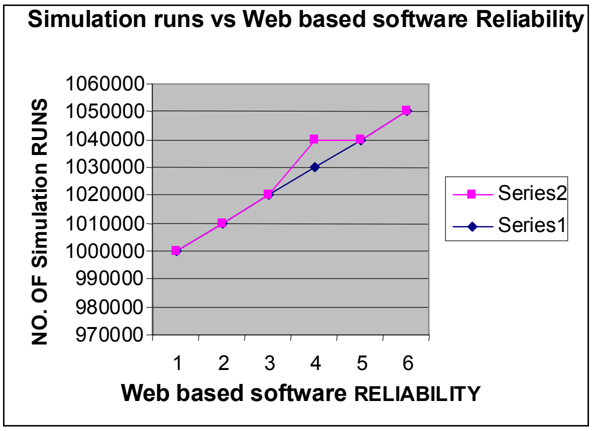
Number of Simulation Runs	Web based software Reliability
1000000	0.947
1010000	0.95
1020000	0.953
1030000	9716
1040000	0.981
1050000	0.99

Table (2)

Description of outputs

The table (2) gives a view of the system reliability computed for different values of Simulation Runs keeping constant values of reliability vector (0.99, 0.98, 0.97, and 0.96). It is found that the system reliability increases as the number of simulation runs increases.

The following graph shows the trend relating to number of simulation runs and the website reliability (System Reliability)



Graph (2)

Discussion & conclusion

The simulator described in the paper will be of great importance to evaluate the reliability of web based software. The transition probabilities of web pages which are not connected directly are considered as zero when

ideal transition probabilities for the web pages are entered Each state is prone to failure to some extent, thereby making the web software imperfect.
The simulator is executed using various combinations of reliabilities of web pages and it is found that the system reliability is sensitive to the reliability of individual web pages. It is depicted from the graph (1) plotted between web page reliability and web based software reliability.
It is also evident from graph (2) that the web based software reliability increases as the number of simulation runs increases using the data shown in table (2).
This simulator will act as an access tool for the software quality assurance team for estimating the reliability of web based system. The reliability evaluation acts as an indicator of amount the software quality assurance needs for meeting the project software quality objectives.

References

[1] Juneja, S. and Sahabuddin, P.,"Efficient simulation of Markov Chains with small transition probabilities," management science, vol 47, no.4, pp.547-562, 2001.

[2] Juneja, S. and Sahabuddin, P., "Fast simulation of Markovian Reliability /Availability models with general repair policies," proceeding of 22nd annual international symposium on fault tolerant computing, IEEE computer society press, pp150-159, 1999.

[3] Davila-Nicanor, L.; Mejia-Alvarez, P. "Reliability evaluation of Web-based software applications" Computer Science, 2005, Sixth Mexican International Conference, P106 – 112, 2005

[4] Jeff Tian, Sunita Rudraraju, Zhao Li, "Evaluating Web Software Reliability Based on Workload and Failure Data Extracted from Server Logs," *IEEE Transactions on Software Engineering*, vol. 30, no. 11, pp. 754-769, Nov 2004

[5] J. Tian, L. Ma, Z. Li, and A.G. Koru, "A Hierarchical Strategy for Testing Web-Based Applications and Ensuring Their Reliability", *1st IEEE International Workshop on Web-Based Systems and Applications* (as part of the *27th Annual International Software and Applications Conference*), Dallas, Texas, Nov., 2003

[6] Li and J. Tian, "Testing the Suitability of Markov Chains as Web Usage Models", *27th Annual International Software and Applications Conference*, Dallas, Texas, Nov, 2003.

[7] Kallepalli and J. Tian, "Usage Measurement for Statistical Web Testing and, Reliability Analysis", *7th International Software Metrics Symposium*, pp.148-158 London, England, April, 2001.

[8] C. Kallepalli and J. Tian, "Measuring and Modeling Usage and Reliability for Statistical Web Testing", *IEEE Trans. on Software Engineering*, Vol. 27, No.11, pp.1023-1036, Nov., 2001.

[9] Tian and L. Ma, "Web Testing for Reliability Improvement", in *Advances in Computers*, Vol. 67, pp.177-224, Academic Press, 2006.

[10] L Ma and J. Tian, "Web Error Classification and Analysis for Reliability Improvement", *Journal of Systems and Software*, Vol.80, No.6, pp.795-804, June, 2007.

[11] Gillette Billy E... "Operations Research", Tata Mc Graw Hill Publishing Company Limited, 2004

[12] Lyu, M.R, and Nikora, A.P.,"Using Software reliability models more effectively," IEEE software, July 1992.

[13] Goel, A.L,"Software Reliability models: A critical survey."IEEE Transactions on Software Engineering, vol .14, no.2, 1982.

[14] Goel, A.L" Software Reliability Models: Assumtion, limitation and application," IEEE transaction on software engineering, vol.11, no. 12, 1985.

[15] Harries, L.N.," the use of models in reliability," in software reliability: Achievement and assessment, B.Little wood, Blackwell Scientific publications, 1987.

[16] Khoshgoftaar, T.M. and Woodcock, T.G., "software Reliability model selection case study", proceeding of 2nd international symposium on software reliability engineering , 1991.

P.K. Suri received his Ph.D. degree from Faculty of Engineering, Kurukshetra University, Kurukshetra, India and Master's degree from Indian Institute of Technology, Roorkee (formerly known as Roorkee University), India. He is working as Professor in the Department of Computer Science & Applications, Kurukshetra University, Kurukshetra - 136119 (Haryana), India since Oct. 1993. He has earlier worked as Reader, Computer Sc. & Applications, at Bhopal University, Bhopal from 1985-90. He has supervised five Ph.D.'s in Computer Science and thirteen students are working under his supervision. He has more than 100 publications in International / National Journals and Conferences. He is recipient of 'THE GEORGE OOMAN MEMORIAL PRIZE' for the year 1991-92 and a RESEARCH AWARD –"The Certificate of Merit – 2000" for the paper entitled ESMD – An Expert System for Medical Diagnosis from INSTITUTION OF ENGINEERS, INDIA. His teaching and research activities include Simulation and Modeling, SQA, Software Reliability, Software testing & Software Engineering processes, Temporal Databases, Ad hoc Networks, Grid Computing , and Biomechanics.

Bharat Bhushan received the M Sc. (Physics), from Panjab Univ. Chandigarh and M.Sc. (Comp. Sc.), MCA degrees from Guru Jambeshwar University. Respectively. Presently working as Head, Department of Computer Science and Applications, Guru Nanak Khalsa College, Yamuna Nagar (affiliated to Kurukshetra University, Kurukshetra-Haryana, India) and senior most teacher of computer science in Haryana since 1984. He is a member of Board of Studies of Computer Science, Kurukshetra University and member of Advisory Board of educational programme (EDUSAT) launched by Govt. of Haryana to impart online education. His research interest includes Software engineering, Digital electronics, networking and Simulation Experiments.