International Journal of Soft Computing 9 (5): 314-317, 2014
ISSN: 1816-9503

Measuring Software Reliability using Fuzzy Logic

M. Ravichandra and A.V. Ramani
Department of Computer Science, Sri Ramakrishna Mission Vidyalaya,
College of Arts and Science, 641020 Coimbatore, India

Abstract: The new generation of technologies is changing the way that customer obtain and construct their Enterprise Computing Systems. Many numbers of end users are using millions of transactions every minute with the help of modular components, data storage devices, high performance networking fabrics, multicore processors and modified software. Hence, software reliability is a key factor in software development process. This study discusses use of fuzzy logic in calculating web software reliability.

Key words: Software reliability, probability, fuzzy logic, membership function, multicore processors

INTRODUCTION

With expanding range of computer applications, computer applications is used more and more in key areas so it requires higher quality and reliability for softwares (Xie *et al*., 2011). At present, the research on software reliability mostly focuses on the software reliability analysis and the study on software reliability problems at extreme conditions is less (Zhu *et al*., 2011). Software reliability is a key factor in software development process. SRGM is a mathematical expression of the software error occurrence and the removal process. Since, the early 1970's many SRGMs have been proposed. A Non-Homogeneous Poisson Process (NHPP) as the stochastic process has been widely used in SRGM. In the past years, several SRGMs based on NHPP which incorporates the Testing-Effort Function conventional (TEF) have been proposed by many researchers. The testing-effort can be represented as the number of CPU hours, the number of executed test cases, etc. Recently, a new SRGM with the Exponentiated Weibull (EW) testing-effort functions to predict the behavior of failure and fault of software is proposed. However, the exponential NHPP growth model is sometimes insufficient and inaccurate to analyze real software failure data for reliability assessment (Raju, 2011).

Fiondella *et al*. (2013) analyzed the reliability of a component-based software system. To support development of more reliable software, US Army Materiel Systems Analysis Activity (AMSAA) has developed a software reliability scorecard offered free of charge to department of defense employees and their contractors. The scorecard methodology provides a structured and transparent approach to assess and improve software reliability practices (Bernreuther and Pohland, 2013). Franco *et al*. (2013) provided relevant information for architects to predict the impact of component reliabilities, usage profile and system structure on the overall reliability. Here, the web software reliability is measured using probability theory and fuzzy logic. The correctness of fuzzy logic for web software reliability measurement is also illustrated.

ERRORS IN SOFTWARE RELIABILITY

The software reliability calculation involves measuring the following three types of errors:

- Database errors
- Hardware errors
- Software errors

Database errors: The purpose of a database is to store and retrieve related information. A database server is the key to information management. A server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. The following are the database errors:

- The data area is full
- The log area is full
- Maximum number of database sessions reached
- The database instance kernel has crashed
- A firewall prevents the connection to an X server on a remote computer
- The X server on a remote computer is not running
- Maximum open cursors exceeded
- Invalid arguments in call

Corresponding Author: M. Ravichandra, Department of Computer Science, Sri Ramakrishna Mission Vidyalaya, College of Arts and Science, 641020 Coimbatore, India

- Invalid column name
- Numeric value out of range
- Data source name not found and no default driver specified

Hardware errors: The hardware errors may be error in local machine, server machine and may be in connection path. These errors are occurred due to failure of electronic component or mechanical component or connecting medium. These types of errors needs replacement of components and require a technical person to handle. Since, these errors are not frequent, technical peoples are called whenever it is necessary. So, it is a time consuming work.

Software errors: Software failures have become a dominant cause of system unavailability. Gray (1990) shows that the main cause of outage has shifted from hardware and maintenance failures to failures in software and to a lesser extent operations. In fact, improvements in hardware and maintenance shrank their contributions to outage from 50-10%. The software errors are register reused, rype mismatch, uninitialized pointer, data error, sequence error, statement logic, synchronization, unclassification, addressing error, incorrect output, infinite loop, etc.

Probability theory: One of the simple measures to access reliability is probability. Here, the errors are classified into three types. Let, x_i, i = 1 to n be the different types of software errors, y_i, i = 1 to n be the different types of hardware errors and z_i, i = 1 to n be the different types of database errors. Let, $p(x_i)$, $p(y_i)$ and $p(z_i)$ are the probabilities of occurrence of software error, hardware error and database error, respectively then reliabilities is calculated as:

$$R = \left[1-\left(\sum p(x_i)+\sum p(y_i)+\sum p(z_i)\right)\right]\times 100 \qquad (1)$$

Since, the sum of probabilities, i.e., $\Sigma p(x_i) + \Sigma p(y_i) + \Sigma p(z_i)$, cannot be >1, the R will give the percentage of reliability of the software.

The probability theory considers all its variables are equal. If all the variables are not equal and it has to be measured on many scales, then the probability theory is no longer useful. The three types of errors software error, hardware error and database error cannot be taken equally for reliability calculations. The maintenance of software, hardware and database differs in terms of its time and effort. The maintenance of database is simple when comparing the other two namely software and hardware.

The technologies changes give more pressure on software and hardware for maintaining it for a larger period. The hardware errors settled one time will not repeat very often but in software the error propagation and time to repair need to be considered. So, instead of using probability it is better to use fuzzy logic for software reliability calculations.

FUZZY LOGIC

Fuzzy logic is conceptually easy to understand, flexible and tolerant of imprecise data, Fuzzy logic can model non-linear functions of arbitrary complexity can be built on top of the experience of experts and based on natural language (Jang and Gulley, 1997). Fuzzy logic is a science of reasoning, thinking and interference that recognizes and uses the real world phenomenon that everything is a matter of degree. It is a computational paradigm that provides a mathematical tool for representing and manipulating information (Yager and Zadeh, 1994). This theory was first raised by the mathematician Zadeh (1965) and it is a powerful modeling technique which has unique ability to capture shades of grey instead of typical black and white approach which has great promise for clinical medicine (Sanchez, 1994).

The calculations of web softwares are complex and uncertain. The fuzzy logic provides not only with a meaningful and powerful representation of measurement of uncertainties, also with a meaningful representation of vague concepts expressed in natural languages. A fuzzy set can be defined mathematically by assigning to easy possible individual in the universe of discourse a value representing its grade of membership in the fuzzy set. This grade corresponds to the degree to which that individual is similar or complicate with the concept represented by fuzzy sets (Fiondella *et al*., 2013).

The web software reliability is measured on three type of errors namely software, hardware and database error. Each error is now calculated as low, medium and high. So, each error needs a membership function. If X's are the different types of software errors $P(X_i)$ is the probability for such error to occur then membership function for software error is:

$$\begin{aligned} S(X_i) &= 0 \text{ if } \sum P(X_i) = 0 \\ &= \sum P(X_i) \text{ if } \sum P(X_i) \le L1 \\ &= 1 \text{ if } \sum P(X_i) > L1 \end{aligned}$$

If Y_i's are the different types of hardware errors $P(Y_i)$ is the probability for such error to occur then membership function for hardware error is:

$$H(Y_i) = 0 \text{ if } \sum P(Y_i) = 0$$
$$= \sum P(Y_i) \text{ if } \sum P(Y_i) \le L2$$
$$= 1 \text{ if } \sum P(Y_i) > L2$$

If Z_i's are the different types of database errors $P(Z_i)$ is the probability for such error to occur then membership function for database error is:

$$D(Z_i) = 0 \text{ if } \sum P(Z_i) = 0$$
$$= \sum P(Z_i) \text{ if } \sum P(Z_i) \le L3$$
$$= 1 \text{ if } \sum P(Z_i) > L3$$

Where:

0 = Error is low
1 = Error is high
$-\Sigma P(X_i)$, $\Sigma P(Y_i)$ and $\Sigma P(Z_i)$ = Medium error
L1-L3 = The tolerance limits for software errors, hardware errors and database errors, respectively which is provided by the user

Now a membership function R for web software reliability can be defined as if $K = A \times S(X_i) + B \times H(Y_i) + C \times D(Z_i)$ then:

$$R = \begin{cases} 1 & \text{if } K = 0 \\ A & \text{if } K > 0 \text{ and } K \le L \\ M & \text{if } K \ge L \text{ and } K \le U \\ 0 & \text{if } K \ge U \end{cases} \quad (2)$$

A-C are weights assigned to errors depending on the influence of different types of errors on software, if the value of R is 1 then software is highly reliable, if it is 0 then is not reliable, if R = A then the software is reliable and if R = M denotes the software can be used for specified period of time. Here, the value of L and M are set depending on the nature of problem.

ANALYSIS OF WEBSITES

The websites of Nailsoft company brindhavan CBSE school and SRMV colleges were taken for analysis. The error occurs in each of the website is given in Table 1. The probability for each type of error for the three companies are given in Table 2.

The theory of probability is used and the reliability is calculated from Eq. 1 as follows: the reliability of Nailsoft Software is $R = (1-0.29) \times 100 = 71$. The reliability of Brindhavan CBSE Software is $R = (1-0.13) \times 100 = 87$. The reliability of SRMV College Software is $R = (1-0.12) \times 100 = 79$. This shows that Nailsoft Software 71% reliable, Brindhavan CBSE Software is 87% reliable and SRMV College Software is 79% reliable.

Table 1: Errors in websites

Errors	Nailsoft	Brindhavan CBSE school	SRMV college
Database error	14	0	0
Hardware error	8	0	4
Software error	10	18	2117
Total number of access	108	134	10,278

Table 2: Probability of errors on different websites

Errors	Nailsoft	Brindhavan CBSE school	SRMV college
Database error	0.13	0.00	0.00
Hardware error	0.07	0.00	0.00
Software error	0.09	0.13	0.21
Total errors	0.29	0.13	0.21

The membership function of database error, hardware error and software error D(X), H(X) and S(X) are also calculated. Here, the value of L is taken as 0.05, i.e., the accepted percentage of error is 5%. Therefore, for nailsoft: S(X) = 0.09, H(Y) = 0.07 and D(Z) = 0.13.

The weightage given for software error is 3 hardware errors is 2 and database error is 1. Therefore: $K = 3(0.09)+2(0.07)+0.13 = 0.54$. From Eq. 2, the value of R is calculated, i.e., R = 0. Similarly for Brindhavan CBSE: S(X) = 0.13, H(Y) = 0 and D(Z) = 0. Therefore, $K = 3 \times 0.13 = 0.39$. This implies that R = 0. Similarly for SRMV college: S(X) = 0.21, H(Y) = 0 and D(Z) = 0. Therefore, $K = 3 \times 0.21 = 0.63$. This implies that R = 0.

The earlier calculations shows that all the web softwares needs corrections or modifications to make it is reliable. Even though probability theory gives positive results for reliability of websites, the results of the fuzzy logic is negative. The reason is probability theory takes all the variables are equal.

CONCLUSION

The web environment has different platforms, different hardware and uses many languages. So, all types of errors cannot be treated equally. Hence, a method using fuzzy logic is proposed for measuring web software reliability and tested with one set of sample data. This shows that fuzzy logic is more suitable for measuring web software reliability. This can be further extended to test the data which will be received from different intervals of time. This needs a method or a technique for consolidating the results from the data with all intervals of times.

REFERENCES

Bernreuther, D. and T. Pohland, 2013. Improved software reliability by application of scorecard reviews. Proceedings of the Reliability and Maintainability Symbosium, January 28-31, 2013, Orlands, Florida.

Fiondella, L., S. Rajasekaran and S.S. Gokhale, 2013. Efficient software reliability analysis with correlated component failures. IEEE Trans. Reliabil., 62: 244-255.

Franco, J.M., R. Barbosa and M. Zenha-Rela, 2013. Reliability analysis of software architecture evolution. Proceedings of the 6th Latin-American Symposium on Dependable Computing, April 1-5, 2013, Rio de Janerio, Brazil Brazil.

Gray, J., 1990. A census of tandem system availability. IEEE Trans. Reliabil., 39: 409-418.

Jang, J.S.R. and N. Gulley, 1997. MATLAB fuzzy logic toolbox. User's guide, Version 1. April 1997, USA. http://www.mty.itesm.mx/dtie/centros/csi/materias/ia4002-1/docs/Fuzzy_Toolbox.pdf.

Raju, O.N., 2011. Software reliability growth models for the safety critical software with imperfect debugging. Int. J. Comput. Sci. Eng., 3: 3019-3026.

Sanchez, E., 1994. Fuzzy logic and inflammatory protein variations. Clin. Chim. Acta, 270: 31-42.

Xie, T., R. Wang, Z. Deng and H. Xiong, 2011. A study on the methods of software testing based on the design models. Proceedings of the 6th International Conference on Computer Science and Education, August 3-5, 2011, Singapore, pp: 111-113.

Yager, R.R. and L.A. Zadeh, 1994. Fuzzy Sets, Neural Networks and Soft Computing. 2nd Edn., Van Nostrand Reinhold, New York., ISBN: 9780442016210, Pages: 440.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353.

Zhu, X.M., Q.Y. Zhang and X. Ren, 2011. The research of software reliability measure based on conditional value at risk. Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology, August 12-14, 2011, Harbin, Heilongjiang, China, pp: 3092-3094.