# Software Reliability Prediction in Component Based System

[1]Gaurav Nagarkoti
M.Tech Scholar, Computer Science and Engineering Department, OM Institute of technology and Management, Juglan (Hisar)- 125001

[2] Saurabh Charaya
Assistant Professor & HOD, Computer Science and Engineering Department, OM Institute of Technology and Management, Juglan (Hisar)-125001

***Abstract*— Traditionally structured programming was used for the software development was simple and successful for small scale software systems. The challenges in software development have been increased with the increasing size and complexity of the software, up gradation after deployment, increased cost and increasing demand of reusable software. Component based software engineering is new paradigm of developing software systems from pre existing components and developing components separately, assembling them into a software system.**

***Keywords*— *Software Reliability, Component based Software Engineering (CBSE), Component Based Systems, Reusability.***

## I. Introduction

CBSE is AN engineering to style the code grouping the parts on the market in repository or by developing new parts as per demand. The CBSE emphasizes on the employ of the element to make a code instead of coming up with it from scratch that makes it totally different from ancient computer code development approaches [1]. The parts that area unit applicable in step with the need area unit chosen from the repository and combined along to develop a desired code as shown in Figure two. the fundamental plan behind the element primarily based system is to employ the parts for building new systems as per the need. A element model could be a that means of gauges for element usages, element Object Model (COM) model, .NET model and customary Object Request Broker design (CORBA) phase Model. The half show indicates however interfaces got to be characterised and therefore the parts that got to be incorporated into AN interface definition [1]. The parts area unit utilised as a district of 2 distinctive settings and 2 sorts: (i) utilizing parts as components to construct a solitary feasible, or (ii) every feasible is restrained as a element during a spread state of affairs, wherever parts team with one another utilizing net or computer network correspondence conventions for IPC (Inter method Communications). the method model for element based programming coming up with stresses parallel tracks during which domain coming up with happens at the same time with element based improvement. Domain building performs the work needed to line up a briefing of programming parts that may be reused by the merchandise engineer. These parts area unit then transported over a "limit" that isolates area building from element primarily based development.

## II. Ease of Use

*Reliability of the Component-Based System(CBS)*

Software dependableness is that the chance of a given system performing arts its task adequately for a such that amount of your time below the expected operative conditions [3]. it's a vital attribute of package quality. dependableness is usually measured as chance of failure, or in terms of convenience, a chance derived from dependableness and maintainability [4]. System with high complexness, together with package, are onerous to succeed in a definite level of dependableness. package dependableness engineering is concentrated on engineering techniques for developing and maintaining package systems whose dependableness are often quantitatively evaluated [5]. part primarily based system area unit designed victimization the off the shelf elements, every part is liable for the dependableness of the system. There area unit the varied factors poignant the dependableness of the system: Operational profile, reusability, maintainability, part dependency, fault tolerance etc [6] instead of this design of the system conjointly the} state transition also plays a vital role to decide the dependableness of the system [1]. The estimation of the dependableness of the system could facilitate within the varied stages of development method.

### *A. Genetic Algorithm*

Reliability prediction of the part based mostly system is completed utilizing factual and numerical techniques as a region of past. These strategies need the precise values or information for analysis. dependableness being planet phenomena changes consequently so creating precise information handiness troublesome. so the soft computing techniques area unit used for the higher results. Soft computing techniques are often accustomed describe planet drawback in higher method [6]. Over the time varied soft computing techniques are used for the analysis of the dependableness.

Genetic algorithms (GA) area unit random search and optimisation algorithms supported the idea of natural process and genetic science. Genetic algorithmic program ideas got in middle sixties by the academic. John Kingdom of The Netherlands of University of Michigan [7]. Genetic Algorithms area unit sensible for big search areas and analyse them for optimum solutions. Genetic algorithmic program required conversion of style house into genetic house. The necessary factors of GA area unit are

- Fitness function definition.
- Definition of variables.
- Implementation of operators

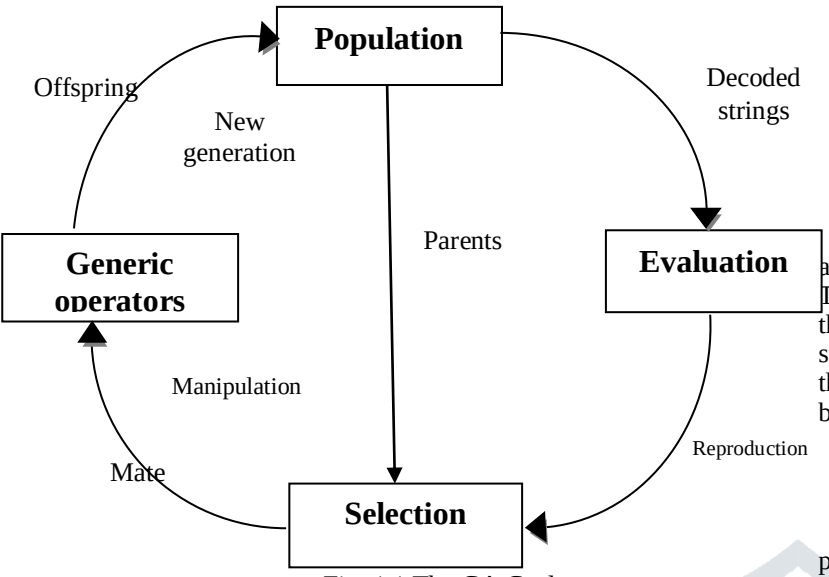


Fig. 1.1 The GA Cycle

LITERATURE REVIEW

Now day's software systems are becoming more complex and large scale that cannot be easily maintained which results in raised maintenance cost, low productivity and complex up gradation [1].

The new approach for the development of the software systems was needed. The object oriented approach however solves the problem to the great extent but was not efficient for the reusability as the objects were difficult to work with. Component based development has effectively reduces the development complexity, increases the reusability and time to market. Component based system is created utilizing the accessible components as per the necessity. The selected components are assembled together to form a software system [8].

2.1 Component

A component is a independent piece of code which is fully functional and is easily deployable.

Component is defined by various researchers as [9]

"A binary unit of independent production, acquisition and deployment that interacts to form a functioning system." - C. Szyperski, Component Software.

2.2 Component Development Lifecycle

Component development lifecycle defines the development phases of component. The development of component must be separated from the system development. The various phases of the development cycle are [11]

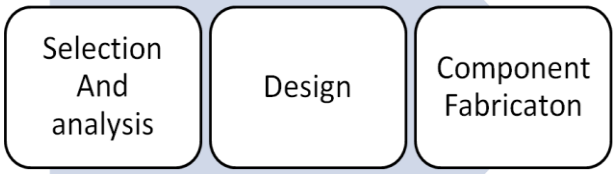


Fig. 1.2 Coponent Development LifeCycle

*2.3 Domain selection and analysis*

Domain selection and analysis is done to indentify the eeds for which the component should be developed. It ecifies the functionality of the component on the basis of e requirement. The selection of the domains for the ailable component must be developed.

2.4 Design of Component architecture

Component based system follow a hierarchical architecture which can be decomposed into smaller levels. The component based system consists of the components and their interfaces. A component contains the algorithm, data structure, controls and interface provides the availability of that component to other component. The compatibility between the two will affect the efficiency of the whole ystem.

2.5 Component fabrication

In this phase component is build and tested. The internal process of the component is only known to the developer. The implementation details are the hide from the user while the component may have many interfaces. The performance and the quality of the components are measured.

2.6 Advantages of CBSE

Following are the advantages of component-based software development as stated by the authors in [1][10]:-

1. It reduces the overall development cost of the software systems as the components can be reused from the repository as and when required.

2. The time to develop the overall software system is reduced to some extent again because components are reusable.

3. The maintainability if the software system is improved as customizing an existing component is much easier than customizing the whole software system.

4. The reliability of the software is increased as testing is performed on each component as well as the whole software system.

5. The up gradation in the software becomes easy.

*3.1 Drawbacks of CBSE*

There are a few difficulties connected with a movement towards a more component based techniques as expressed as in [11].

- There is an expansion in multifaceted nature in the product connected with the extended utilization of libraries. Generally speaking, staff will require enhanced programming building and outline abilities.
- Projects will have an expanded reliance on library engineers, who may not be a piece of the venture and be liable to contending needs.
- Componentization can prompt decreased execution because of the need to go information through extra interfaces furthermore in light of the fact that general and compact calculations will most likely be unable to exploit unique structure in a particular issue.

3.2 Current Component Technologies

The component-based software development relies on certain technologies or underlying services to provide the communication and co-ordination necessary for the application. These technologies are described as follows [1] :-

3.2.1 Common Object Request Broker Architecture (CORBA)

Common Object Request Broker Architecture (CORBA) is gotten by the Object Management Group (OMG) it gives experiences about the interfaces and characteristics of the parts of the Object Management Architecture and particular for making, conveying, and overseeing appropriated program objects in a system. It permits programs at various areas and created by various sellers to convey in a system through an interface merchant [12]. The CORBA enables the interoperability and communication between the components despite of their location and design. The main part of the CORBA is Object Request Broker. The object request broker helps in establishing the client server relations between the components [1]. CORBA is widely used in component based systems because it provides an independent platform for runtime environment.

3.2.2 Common Object Model (COM) AND Distributed COM (DCOM)

Common Object Model (COM) determines fundamental segment programming engineering. It gives stage subordinate window based part programming frameworks [1]. COM defines the interaction between the components either within the same process or in different processes. The interaction between the components over a network is defined by Distributed Component Object Model (DCOM).

DCOM (Distributed Component Object Model) is an arrangement of Microsoft ideas and system interfaces in which customer program items can ask for administrations from server program objects on other computers in a network[13][14]. DCOM provides the multiple network transports, when clients and their components reside in different machines; DCOM replaces the inter process communication with the system convention.

3.2.3 Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) is a development model for client and server side components. It supports the multiple platforms applications and client side and server side components. It uses java programming for the development thus provides the platform independent and provides effective solution for portability of the components and security [1].

EJB components works in a container and each component interacts using an interface which provides creation, migration and eradication. The components in the complaint containers can freely move from one container to another without any reconfiguration [15]. EJB components are easily portable, platform independent and can be maintained and modified easily.

4.1 Related work

Reliability of the software has always been a keen interest area for the researchers. Many models have been proposed and can be categorized as [16]: architecture based models, mathematical based models and soft computing techniques.

Design based models makes a note of programming engineering and operational profiles of the product framework to model its reliability. Popstojanova and Trivedi [17] said that state based models uses markov chains and markov methodology to portray the outline of the item and join the failure rate with engineering to foresee the dependency of the system. The path based models uses the diverse execution routes for figuring the dependability.

Wang and Dai Pan [18] familiar a configuration based model with appraisal programming relaibility in any phase without any requirement of test data with the help of markov model.

Wang and Ye Wu [19] proposed a model for reliability estimation which take in assessment change in the state of the structure close by the components and the outline styles.

Tyagi and Arun Sharma [20] proposed a model in view of heuristic components reliance diagram utilizing ant colony optimization. This model maps the execution ways that are generally utilized and along these lines evaluates the dependability of the parts followed in that way.

Anurag and Pradeep Tomar [21] proposed a model that maps the propagation of paths and component's effect on the system. The model portrays the impact of the active components amid the way spread and demonstrates the components utilized all the more amid execution greatly affect the reliability of the software system.

Heiko and Becker [22] introduced a model which accepts the operational profile of the frameworks, and utilizes the change of the operational profiles to show the framework reliability.

Goswami and Acharya [23] proposed a methodology which utilizes numerical equations to figure component utilization proportion which is the execution time of the segments to the general execution time. This methodology can be utilized to assess the general dependability of the framework.

Yacoub and Bojan [24] proposed a model, depends upon the execution case of the structure using the Component Dependency Graph . A count is described in light of CDG which uses the unwavering quality of the parts and interfaces as the limit for examining the reliability of the application.

Brosch and Franz [25] proposed a model in perspective of the Palladio Component Model, and utilize a parameterized UML-like model for surveying auxiliary design choices. It licenses influence capacity examination of the structure dependability to the probability of individual disappointment, changes in system level usage profile and changing gear openness if there ought to emerge an event of disappointment. It focuses on the inducing of the usage profile all through the designing, and the execution environment influences on the application.

Si et al. [26] introduced a system for assessing dependability by a part organization mechanism. The methodology uses five essential part organization mechanism and strategies for their reliability estimation. Subsequent to computing the unwavering quality for every synthesis, a technique gauges the general application reliability in light of the part arrangement mechanism and component use frequencies. It is conceivable to perceive extra organization mechanism.

Shukla et al. [27] introduced an orderly strategy which gave a regulated methodology for creating operational profiles for programming parts. The technique has utilized both usage information and planned use assumptions to find a use appropriation, use structure and attributes of parameters.

The probabilities of the operations are characterized by utilization appropriation. The technique built up an utilization structure utilizing rules from the data assembled in the initial step, and afterward displayed the use structure utilizing state graphs. The utilization quantification process delivered a use conveyance. The parameter investigation process characterized parameter qualities. To show the technique, creator connected it to the Symbol Table. The strategy was proposed to be a piece of a more broad structure for the reliability evaluation of programming parts. The structure has included backing for experiment execution and yield assessment.

Gayen and Misra [28] proposed a creative way to deal with anticipate the upper and lower bound on the dependability of the COTS Component-Based programming application. On the premise of the execution situation investigation for the COTS Component-Based Software System, an unmistakable strategy was figured. The proposed calculation for the expectation of upper bound was a change over Dolbec and Shephard [29] model for estimation of dependency of Component-Based Software. The disadvantage of his model was that it was execution way free and segment interfacing time was not considered. Along these lines, it was not able anticipate the upper bound on dependability, as the upper bound on unwavering quality acquired utilizing Dolbec and Shephard [29] model was considerably less than the worth got in the proposed approach. The proposed methodology was pertinent under any preparing environment be it bunch or parallel handling in a uni-processor or a multiprocessor framework. Since, in this approach one was for the most part worried with the segment utilization proportion and not with the clock time for the execution of the segment. Thus, it was expected that the designer was a perfect engineer who codes effectively (as the dependability of the interfacing code was thought to be solidarity) interfacing the COTS segments with no blunder (which may later on cause disappointment of the application framework) to deliver the item which may not generally be in actuality.

Singh et al. [30] exhibited a Bayesian reliability estimation model utilizing a bound together demonstrating dialect (UML) strategy for dependability forecast and evaluation. The method gives unwavering quality investigation at the outline level, i.e., before the framework improvement and incorporation level. Since this methodology depends on UML graphs, re-risk can be anticipated in the early plan stage. This methodology is versatile on the grounds that all figuring are finished by a mechanized apparatus. The methodology has one impediment: If new ways are considered, the unwavering quality pre-lingual authority calculation considers this to be another framework, so that different operational profiles are created.

Fiondella et al. [31] proposed a methodology taking into account corresponded component failures. In this paper a proficient way to deal with access the unwavering quality of a product application, considering the component reliabilities, relationship and application engineering is proposed. This proposed methodology depends on a calculation that changes Multivariate Burnoulli dispersion (MVB) into a joint dissemination of the segment results. The viability of the methodology was exhibited through two trials, as a result of its effectiveness, the methodology is additionally reasonable for dissecting the affectability of the application dependability to segment to relationship parameters. This methodology is straightforward and effective. The methodology might be connected to vast frameworks to recognize co-relations that block framework dependability.

Wang and Huang [32] proposed a methodology for dependability estimation taking into account revise rationale. This strategy considers frameworks whose particular is given with an operational profile. Maude's rework method is utilized to appraise the unwavering quality. This procedure factually dissects an application's execution procedure and utilizations this to around evaluation the move probabilities amongst components and the normal number of visits to components. In any case, this methodology has a few confinements: First, it must be connected to straightforward CBSs, and second, it doesn't consider disappointment conditions between components.

Numerous strategies have been acquainted with appraisal the reliability of the component based framework, new drifting methodologies are soft computing methods. Past methodologies utilizes the exact information for the examination while the soft computing procedures are reasonable for the imprecise information and work best for this present reality issues. Since reliability of the software's changes as needs be in this manner precision can be accomplished utilizing these procedures.

Lo et al .[33] proposed a product dependability estimation model in view of a state vector machine and a genetic calculation (GA). This model accepts that late disappointment information alone are adequate for assessing dependability. Reliability estimation parameters for the SVM are dictated by the GA. This model is less subject to failure information than are different models.

Dimov and Punnekkat [34] introduced a model for component based framework using the fuzzy logics which utilizes the vulnerability as a part of the framework and consequently does not require component failure information. It depends on fuzzy rationale and possibility hypothesis. This model requires a procedure to characterize the engendering of failure among the components.

Proposed Meathodology

5.1 Genetic Algorithm

Genetic algorithm is introduced by professor john Holland in 1975, based on the genetic evolution theory of Darwin. It uses rules of natural selection to solve the problems. It is a random search algorithm. It performs operation on the expected population of solutions on the basis of survival of the fittest for better approximations to the solution. At each generation best individuals are selected on the basis of their fitness in that environment and breeding them to generate a new population with best fitness for the provided environment [7].

Individuals and approximations are encoded as strings which are known as chromosomes in natural selection. The coded strings are called genotype which is mapped to the decoded strings which are called phenotype [36].

| Natural Selection | Genetic Algorithm |
|---|---|
| Chromosomes | Strings |
| Gene | Features |
| Allele | Values of features |
| Locus | Position of String |

| | |
|---|---|
| Genotype | Coded String |
| Phenotype | Decoded String |

Table 5.1 Difference between Natural selection and Genetic algorithm terminology[7]

A simple genetic algorithm is consists of following steps [7]:

1.Selection: This step is done for the selection of best individual according to their fitness for reproduction. Selection can be done by various methods.

2.Reproduction: The selected individuals are mated to generate off springs either by mutation or crossover.

3.Evaluation: The new population is evaluated for the fitness.

4.Replacement: Old population is replaced with the new population.

The genetic algorithm stops when the optimal solution is reached.
The basic genetic algorithm steps are explained in Figure 6. In the first step the population is selected randomly on the basis of type of selection. Each individual is computed for its fitness. Then the best individuals are stored.

In second step the selected individuals are bred to form off springs or next generation using crossover. Crossover may be single point crossover or two point or multiple point crossover. Crossover is done to get new population, otherwise the new population will have same genes as that of parent population.
If the optimal solution is obtained the genetic algorithm is terminated otherwise again few population is ignored and is reproduced again.
Mutation is done to new generated population. The positions in the chromosomes are mutated according to the mutation rate.
The new generation obtained is replaced with the old generation and evaluated for the fitness.

Difference between the traditional optimization techniques and Genetic algorithm

1) GA works with coded variants of the issue parameters as opposed to parameters themselves i.e., GA works with the coding of arrangement set and not with the arrangement itself.

2) Almost all ordinary streamlining strategies seek from a solitary point however GAs dependably works on an entire population of points (strings) i.e., GA utilizes population of arrangements as opposed to a solitary arrangement for looking. This assumes a noteworthy part to the strength of hereditary calculations. It enhances the possibility of achieving the worldwide ideal furthermore helps in keeping away from nearby stationary point.

3) GA utilizations, fitness capacity for assessment as opposed to subordinates. Subsequently, they can be connected to any sort of consistent or discrete advancement issue. The key point to be performed here is to recognize and indicate an important unraveling capacity.

4) GAs use probabilistic move works while ordinary strategies for constant streamlining apply deterministic move works i.e., GAs does not utilize deterministic tenets.

Advantages of GA

1) GA provides parallelism and liability.

2) Solution space is large, handles complex search spaces easily.

3) Use of modulation of functions.

4) Provides global optimal solution.

5) The fitness scene is unpredictable.

6) Simple to find worldwide ideal.

7) Effectively modified for various issues.

8) Handles substantial, inadequately comprehended hunt spaces effectively.

9) Useful for multi-modular issues, Returns a suite of arrangements.

10) Extremely vigorous to difficulties in the assessment of the goal capacity.

Limitation of GA

1) The issues of recognizing fitness function.

2) Definition of representation for the issue.

3) Untimely convergence occurs.

4) The issue of picking the different parameters likes the measure of the population, change rate, traverse rate, the determination technique and its quality.

5) Can't utilize inclinations.

6) Can't without much of a stretch use issue specific data

7) Not great at recognizing nearby optima

8) No powerful eliminator.

9) Not powerful for smooth uni modal capacities

10) Should be combined with a nearby hunt system.

11) Experience difficulty in finding the precise worldwide ideal.

12) Require huge number of reaction (fitness) capacity assessments.

13) Configuration is not direct.

5.2 Genetic Algorithm using Optimization Tool

The genetic algorithm can be access using the MATLAB. The MATLAB provides optimization of genetic algorithm at command line as well as using optimization tool.

To use genetic algorithm at command line

At command line genetic algorithm can be called using the function ‘ga’[37].

Syntax of function ga

[x, fval] = ga (@fitness_fun, nvars, options)

where

@fitness_fun = handles the objective function(fitness function)

nvars = number of variables

options = is used to specify genetic algorithm options.

The output is obtained as:

x = is the point where final value is obtained

fval = final value of the objective function

To use genetic algorithm by optimization tool

To open optimization tool the command "optimtool" is used. Then the Genetic algorithm solver is opened.

To use optimization tool the following data must be entered [37]:

Fitness function – The fitness function is the objective function which is being optimized. Fitness function is called using @fitness which calls the fitness.m file containing the objective function.

Number of variables – the number of variables used by the fitness function must be passed to the solver.

Other parameters are optional such as

Linear constraints on the objective function.

Non linear constraints on the objective function.

Lower bound and Upper bound on the variables.

*6.1 Objectives*

Following are the objectives that are achieved in this thesis:-

- Discussion over the various methods has been proposed for the reliability estimation of the component based system.
- Proposing a model based on genetic algorithm for predicting reliability on the basis of the transitions between the components.
- Validation

7.1 Conclusion

Component based software engineering is emerging as new trend to overcome the challenge of maintaining the complex systems. It promotes the reuse of the components rather than implementing the whole system. Components can run independently if properly designed with less dependency on the environment. It requires quality assurance for the components to increase the productivity and efficiency of the software systems. It is an effective and easy method for developing software system in short time with lesser cost. While making any changes in software, it would be possible to predict certain reliability factors to check the compatibility with the results of the change. Thus this model predicts the reliability of the system using the transition probabilities and component reliabilities. Use of GA provides the optimal solution in less computation time and any type of changes required in the procedure can be done manipulating the operators in genetic algorithm.

REFERENCES

[1] Cai, Xia, et al. "Component-based software engineering: technologies, development frameworks, and quality assurance schemes." Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific. IEEE, 2000.

[2] Crnkovic, Ivica, Michel Chaudron, and Stig Larsson. "Component-based development process and component lifecycle." Software Engineering Advances, International Conference on. IEEE, 2006.

[3] "Ning, Jim Q. "Component-based software engineering (CBSE)." Assessment of Software Tools and Technologies"., pp 34-43 IEEE, 1997.

[4] Upasna Jaglan Deepesh Shrivastava "A straight map of Software Reliability on component based Engineering" , Volume 2, Issue 3, IJARCSSE 2012.

[5] Lyu, Michael R. "Software Reliability Engineering: A Roadmap." FOSE 7 (2007): pp 153-170.IEEE 2007.

.

[6] Kirti Tyagi, Arun Sharma "Significant Factors for Reliability Estimation of Component Based Software Systems", Journal of Software Engineering and Applications,pp 934-942, 2014

[7] Sivanandam, S. N., and S. N. Deepa. Introduction to genetic algorithms. Springer Science & Business Media, 2007.

[8] Breivold, Hongyu Pei, and Magnus Larsson. "Component-based and service-oriented software engineering: Key concepts and principles." Software Engineering and Advanced Applications, 2007. 33rd EUROMICRO Conference on. IEEE, 2007..

[9] Chaudron, Michel, and Ivica Crnkovic. "Component-based software engineering." Software Engineering: Principles and Practice (2008): pp 605-628 2008.

[10] Christiansson, Benneth, and Lars Jakobsson. "Component-Based Software Development Life Cycles." Karlstad Universitet, Institutionen för Informationsteknologi 1999.

[11] Sharp, Jason H., and Sherry D. Ryan. "Component-Based Software Development: Life Cycles and Design Science-Based Recommendations."Proc CONISAR, v2 (Washington DC) 2009.

[12] Vinoski, Steve. "CORBA: integrating diverse applications within distributed heterogeneous environments." Communications Magazine, IEEE 35.2 (1997): pp 46-55 1997.

[13] "A Systematic Approach for Component-Based Software Development" by Cléver Ricardo Guareis de Farias, Marten van Sinderen and Luís Ferreira Pires, Centre for Telematics and Information Technology (CTIT), 2001.

[14] "A Tool for Component Based Design of Embedded Software" by David Urting, Yolande Berbers, Stefan Van Baelen, Tom Holvoet, Yves Vandewoude, Peter Rigole, Department of Computer Science, Catholic University of Leuven, 2002.

[15] Emmerich, Wolfgang, and Nima Kaveh. "Component technologies: Java beans, COM, CORBA, RMI, EJB and the CORBA component model."Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on. IEEE, 2002.

[16] Tyagi, Kirti, and Arun Sharma. "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems." Applied Computing and Informatics pp 38-51 2014.

[17] Goseva-Popstojanova, Katerina, and Kishor Trivedi. "Architecture based software reliability." 2000.

[18] Wang, Wen-Li, Dai Pan, and Mei-Hwa Chen. "Architecture-based software reliability modeling." Journal of Systems and Software pp 132-146, 2006.

[19] Wang, Wen-Li, Ye Wu, and Mei-Hwa Chen. "An architecture-based software reliability model." Dependable Computing, 1999. Proceedings. 1999 Pacific Rim International Symposium on. IEEE, 1999.

[20] Tyagi, Kirti, and Arun Sharma. "A Heuristic Model for Estimating Component-Based Software System Reliability Using Ant Colony Optimization." World Applied Sciences Journal vol 31 issue 11 pp 1983-1991,2014.

[21] Singh, Anurag Prakash, and Pradeep Tomar. "A New Model for Reliability Estimation of Component-Based Software." Advance Computing Conference (IACC), 2013 IEEE 3rd International. IEEE, 2013.

[22] Koziolek, Heiko, and Steffen Becker. "Transforming operational profiles of software components for quality of service predictions." Proceedings of Workshop on Component Oriented Programming (WCOP). 2005.

[23] Goswami, Vivek, and Y. B. Acharya. "Method for reliability estimation of COTS components based software systems." International Symposium on Software Reliability Engineering. 2009.

[24] Yacoub, Sherif, Bojan Cukic, and Hany H. Ammar. "A scenario-based reliability analysis approach for component-based software." Reliability, IEEE Transactions on vol 53 issue 4 pp 465-480, 2004.

[25] Brosch, Franz, et al. "Parameterized reliability prediction for component-based software architectures." Research into Practice–Reality and Gaps. Springer Berlin Heidelberg, pp. 36-51, 2010.

[26] Si, Y.J., Yang, X.H., Wang, X.Y., Huang, C. and Kavs, A.J. (2011) An Architecture-Based Reliability Estimation Frame- work through Component Composition Mechanisms. 2nd International Conference on Computer Engineering and Technology, Chengdu, pp 16-18 April 2010, pp 165-170, 2011.

[27] Shukla, R., Carrington, D. and Strooper, P. (2004) Systematic Operational Profile Development for Software Compo- nents. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, , pp 528-537, 2004.

[28] Gayen, T. and Misra, R.B. Reliability Bounds Prediction of COTS Component Based Software Application. IJCSNS International Journal of Computer Science and Network Security, 8, pp 219-228, 2008.

[29] Dolbec, J. and Shepard, T. (1995) A Component Based Software Reliability Model. Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research, Toronto, pp 7-9 1995.

[30] Singh, H., Cortellessa, V., Cukic, B., Gunel, E. and Bharadwaj, V. (2001) A Bayesian Approach to Reliability Prediction and Assessment of Component Based Systems. 12th IEEE International. Symposium on Software Reliability Engineering, Hong Kong, pp 12-21,2001.

[31] Fiondella, L., Rajasekaran, S. and Gokhale, S. (2013) Efficient Software Reliability Analysis with Correlated Compo- nent Failures. IEEE Transaction on Reliability , pp 244-255, 2013.

[32] Wang, D. and Huang, N. (2008) Reliability Analysis of Component Based Software Based on Rewrite Logic. 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, pp 126-132, 2008.

[33] Lo, J.-H. (2010) Early Software Reliability Prediction Based on Support Vector Machines with Genetic Algorithms. International Conference on Industrial Electronics and Application, pp 2221-2226, 2010.

[34] Dimov, Aleksandar, and Sasikumar Punnekkat. "Fuzzy reliability model for component-based software systems." Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on. IEEE, 2010.

[35] Siddiqui, Z., and Kirti Tyagi. "Application of fuzzy-MOORA method: Ranking of components for reliability estimation of component-based software systems." Decision Science Letters vol 5 issue 1 pp 169-188 2016.

[36] Chipperfield, A. J., and P. J. Fleming. "The MATLAB genetic algorithm toolbox." Applied control techniques using MATLAB, IEE Colloquium on. IET, 1995.

[37]MATLAB(2013) http://in.mathworks.com/help/gads/genetic-algorithm.html .The Mathworks.inc,.USA, 2013.