

A Model based Approach for Evaluating the Reliability of the Services in the Service-oriented Architecture

Behzad Alipour*, Ali Haroon Abadi**

* Islamic Azad University Ilam Science and Research Branch

** Islamic Azad University Central Tehran Branch

Article Info

Article history:

Received Jun 12th, 2015

Revised Aug 20th, 2015

Accepted Aug 26th, 2015

Keyword:

Service-oriented architecture

Reliability

Markov chains

Web services

ABSTRACT

Service-oriented architecture presents a frame in which the system functions are defined as a series of the distributed services in the intended sizes of the organization. These services are called by the other software and also are used for building the new services. Although this architecture offers a simple solution for building the distributed systems with loosely coupling, it introduces some additional concerns. One of the main concerns in designing a SOA system is general reliability of the system. Then the new technique for modeling the reliability is needed for certificating the services. Regarding to this weakness, in this paper, a certification method for the reliability in which services have been simulated as the discrete Markov chains, this work presents a model for estimating the reliability by exploiting the mean and variance of the visits obtained from analyzing Markov chain and integrating them into the reliability of the characteristics of each individual service. Results show that in the proposed method, less fault-tolerance than the recent methods for predicting the reliability of the systems is used.

Copyright © 2016 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Bezad Alipour,

Department of Electrical and Computer Engineering,

Islamic Azad University, Science and Research Branch, Ilam, Iran.

Email: alipour.behzad@yahoo.com

1. INTRODUCTION

Service oriented architecture (SOA) is a promising computational pattern for developing the software in an open heterogeneous environment [1, 2]. Definition of SOA in the perspective of IBM is an approach for building the distributed systems which present the software functions in the service form. These services are callable by the other software and also are used for building the new services [3]. Also increasing demand for flexibility and scaling in integrating and reusing the software has resulted in the wide acceptance of the web services and SOA functional programs. In this architecture, wide spectrums of the services distributed by different providers have been interactively integrated with XML-based web services protocols. A digital ecosystem allows forming the design and orchestration of the business process [5, 4]. Despite some advantages, this dynamic pattern for software building has changed the amplitude and dimensions of the risks in developing the business process. As a result, users are increasingly concerned with the failures affecting the functional and non-functional properties of their programs. In this respect, reliability of the services is considered as an important factor by the users and increases the need to SOA-based confirmation, credit and reliability techniques [6-8]. Defining the techniques for guarantee the increasing trust of the users is very important. While increasing the trust of the users, the services should be in agreement to the needs of the reliability. Main aim of this study is to present a method for predicting the reliability of the services. In the service-oriented systems, services, integration of the services and input/output data of the service are separated; they may be provided, presented and stored by the different providers and are used by the different functional programmers for building the system in the various areas of the business process. Reliability of a

SOA system is depended on the reliability of its services and the input data from the external source. However, different providers may provide the different qualities of the service.

For solving this problem of service-based systems, new methods and modelling techniques are required. Until now, limited efforts have been done in this area [9-12]. In this paper, we present a hierarchy method for modeling and analyzing the reliability of the services. First step in this method is modeling the service by using Discrete Time Markov Chains [13]. Specially, we use Markov model to guarantee that the service supports the given reliability with the given reliability level. Then, we will present a new method for estimating the reliability of the systems by using the mean and variance of the visits obtained from Markov chain analysis and integrating them into the reliability of the properties of individual services. In this paper, an example has been simulated for explaining the proposed method and comparing it with the other recent methods.

Relating the solved problem in this paper, Wang et al have proposed a hierarchy reliability method in which reliability of the system is computed by placing the credit layer on the data, services, fault tolerance mechanism and the logic of the service combination [9]. Claudio et al presented a certification plan for reliability in which the services are modeled as discrete time Markov chain (DTMC); their certification accuracy is affirmed by monitoring the constant implementation time. Moreover, they presented a solution which allows the users to search and choose their services by a series of the given reliability properties [10]. Delac et al, focused on the shift toward predicting the fault and the techniques for fault tolerance. They presented a model for estimating the reliability of a complex service by using the graph which is used for the complex services based on the orchestration model [14]. Rahmani et al analyzed the reliability of the service-based software by using the internal and external services and used the Petri network model for evaluating the reliability of the composite software [15]. Grassi et al, worked on the measuring the quality of the reliability of the systems and presented an implementation of the methodology for the reliability by considering the independency and non-concentration of the services [23]. In this paper, we have tried to improve the results of the past researchers.

1.1. Reliability of the Service

An action which is done by a service provider and is valuable in his/her opinion may be doing a small request such as receipt and storing the data or be related to a more complex function such as processing an image. In the SOA, service is usually defined more specially and is related to the object and the parameter. It is clear that there is no necessity for using the object-oriented methodologies and as mentioned before, SOA is independent from the platform; but it can be said that the concept of "service" has some similarities to the object and the parameter.

Reliability is a special aspect of the wider concept trust. Other aspects of reliability and trust are for example availability and safety. Reliability specially refers to the continuance of the services provided by a system. In this respect, two basic definitions of the reliability can be found in the history of the research [16]. First definition is probability of doing its required functions under the mentioned conditions for a determined period by the system. Second definition is probability of doing successfully the function if requested. First definition refers specially to not-ending forever unless the system performs correctly the delegated duty in the determined time (for example in the flight control system of a plane in which no failure is permitted). Second definition refers to the service provider systems that when are requested, they will be ended successfully. It should be noted that the focus of this study is on the second definition.

1.2. Discrete time Markov Chain (DTMC)

In this section, DTMC that is used for modeling a web-based system is discussed. Markov process is a random process in which dynamic behavior of probability distribution for the future developments depended on the present state not on the manner of transferring to this state. But if we suppose that the state space I and also the space parameter T are discrete, then a Markov chain will be valued by its states and the probability of the transmissions among the states.

The transition probability matrix $P = [p_{i,j}]$ is a random matrix as all the components in a row of P add up to 1 and each of the $p_{i,j}$'s is in the range of 0~1.

For our goal, we could categorize the DTMC into two classes [17]: 1) Irreducible, If every state can be reached from every other state in a specific number of steps. 2) Absorbing, that correspond to each state can be available for the other states in the limited steps.

Matrix P is a simplifier DTMC which can be divided as the following:

$$P = \begin{bmatrix} Q & C \\ 0 & 1 \end{bmatrix} \quad (1)$$

If Markov chain has n states with m absorbing states, element 1,1 of the matrix P is itself a sub-stochastic matrix $(n-m) \times (n-m)$ which defines the probability of the transmission among the unstable states, element 2,2 of the matrix P is itself a single $m \times m$ matrix which shows the probability of the transmission of each m states to itself, element 1,2 of the matrix P is itself a $m \times (n-m)$ matrix having zero value and element 2,1 of the matrix P is itself a $(n-m) \times m$ matrix. For m reduced states of this matrix is ignored, matrix of probability of the k -steps transmission is shown as P^k :

$$P^k = \begin{bmatrix} Q^k & C^k \\ 0 & 1 \end{bmatrix} \quad (2)$$

The (i, j) Th entry of Q^k shows the probability of the transmission from the state S_i to S_j . It can be shown that $\sum_{k=0}^t Q^k$ is diverged to t , then the reverse matrix of $(I-Q)^{-1}$ existed. Then basic matrix M is considered as the following [18]:

$$M = I + Q + Q^2 + Q^3 \dots = \sum_{k=0}^{\infty} Q^k = (I-Q)^{-1} \quad (3)$$

If X_{ij} is the numbers of the visits of the state j which has been given in (i, j) Th entry of the basic matrix M , then I is an identity matrix having the same dimensions as Q . where m_{ij} is the mathematics promise of X_{ij} . Then we have [19, 23]:

$$E[X_{ij}] = m_{ij} \quad 1 \leq i, j < n \quad (4)$$

Variance of the expected visits can be calculated by using the basic matrix. If $\sigma_{i,j}^2$ is the numbers of the routes from s_i to s_j , the variance is obtained as the following:

$$md_{i,j} = \begin{cases} m_{i,j} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In the other word M_D presents a diagonal matrix having M diagonal inputs. We have [20, 23]:

$$\sigma^2 = M(2M_D - I) - M_2 \quad (6)$$

Then:

$$\text{Var}[X_{i,j}] = \sigma_{i,j}^2 \quad (7)$$

For a program comprising some software services, we can represent our architecture by using a DTMC. One state of Markov chain shows the implementation of a service. Moreover, transmission among the states of Markov chain showing the transmission of the control from one service to another is presented.

2. MODELING THE SERVICE

Modeling the service as finite state automata has been previously used for estimating and improving the reliability and also testing and affirming the security properties of the service. Issuing the presented certification in this paper is by using the state-based DTMC which integrates the failure behavior and architecture of the service for affirming a given series of the reliability properties. We have shown the circuit control of a service as a graph $G = (N, E)$ in which each node $n \in N$ is related to the service state and a directional edge $(n_i, n_j) \in E$ shows the transmission of the control from n_i node to node n_j . Transmission from one state to the other is assumed as a Markov process without considering the point of the time in which the transmission is occurred.

Graph G has been enriched with a) policy conditions related to each state transmission and b) states C and F which represent the correct output and failure, respectively. When service satisfies the policy, state C is attainable and also in the case of the failure or defect of the policy, it attains at F state. Probability of transmission of $Pr_{i,j}$ is related to satisfying the condition of the policy with each directional edge (n_i, n_j) and

R_i remains as the probability of each free failure related to each node n_i . In this paper, $R_i Pr_{i,j}$ shows the probability of implementing a service which produces the accurate result in N_i state and transmits the control to state n_j . Passing from the final state n_k to state C yields the probability R_k . It is seen that the service satisfies all the conditions in *pol* (without failure). Note that presence an implicit failure in the probability statement

$1 - \sum_{j=1}^k R_i Pr_{ij}$ from each node $n_i \neq n_k$ toward F shows a failure or breach the conditions of the policy.

Transmission from n_k to F yields the probability $1 - R_k$. Induction model from graph G is defined as the following [10]:

Service model has seven taples $\{n_1, N_1, n_k, C, F, \xrightarrow{c_{ij}}, R_i Pr_{ij}\}$ where n_i is an elementary state. Shows all middle states $n_2 \dots n_{k-1}$. And n_k is the final state of the complementation. C is the correct final state, F is the final failure state $\xrightarrow{c_{ij}}$ shows the transmission between two node (n_i, n_j) with the policy condition tag $c_{ij} \in Pol(p)$, $R_i Pr_{i,j}$ shows the probability of yielding the correct result by implementing a service and satisfying the policy conditions in n_i and transmission to n_j state. Defined model as Q' is shown in the following [10]:

$$Q' = \begin{matrix} & \begin{matrix} C & F & n_1 & n_2 & \dots & n_k \end{matrix} \\ \begin{matrix} C \\ F \\ n_1 \\ n_2 \\ \vdots \\ n_k \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 - \sum_{j=1}^k R_1 Pr_{1j} & R_1 Pr_{11} & R_1 Pr_{12} & \dots & R_1 Pr_{1k} \\ 0 & 1 - \sum_{j=1}^k R_2 Pr_{2j} & R_2 Pr_{21} & R_2 Pr_{22} & \dots & R_2 Pr_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ R_k & 1 - R_k & 0 & 0 & \dots & 0 \end{bmatrix} \end{matrix} \quad (8)$$

Note that for more transparency, policy conditions haven't been reported in Q' . Following the presented approach in [18], affirmation licence uses matrix Q' to estimate which service satisfies the reliability of p . the matrix Q is obtained from matrix Q' with deleting rows and the columns corresponding to C and F . Then we consider matrix M as the relation (3), so that in matrix M , I is a single matrix having the same dimensions as Q . Therefore, reliability level of the service is defined as the following.

2.1. Prediction of the reliability

Reliability of a SOA composes of some software services which can be calculated by using the mean and variance of the visits obtained from analyzing Markov chain and integrating it into the reliability of the individual service [22]. Assume that s_i is i^{th} state of the implementing service, suppose that failure and success of the service states are independent. If R_i is reliability of service I and $x_{i,j}$ is the number of visits s_j starting from s_i , then general reliability of a service with regarding to this fact that all service states should correctly operate to guarantee the reliability of the service is given as:

$$R = \prod_{i=1}^n R_i^{x_{i,i}} \quad (9)$$

Note that the numbers of the visits to each service state is itself a random variable, R is also a random variable. Expected reliability of the random variable R is estimated as the following:

$$E[R] = E\left[\prod_{i=1}^n R_i^{x_{i,i}}\right] = \prod_{i=1}^n E[R_i^{x_{i,i}}] \quad (10)$$

Here, $R_i^{x_{i,i}}$ is the expected reliability of state I in a single implementation. By using first tridominal estimation of Taylor series extension, we have [24]:

$$f(x)=f(x_0)+\frac{f'(x_0)(x-x_0)}{1!}+\frac{f''(x_0)(x-x_0)^2}{2!} \tag{11}$$

Then, substituting the corresponding sentences we have:

$$E[R_1^{x_{1,i}}]=R_1^{E[x_{1,i}]}+\frac{1}{2}(R_1^{E[x_{1,i}]})^2\text{Var}[x_{1,i}] \tag{12}$$

This can be written as:

$$E[R_1^{x_{1,i}}]=R_1^{m_{1,i}}+\frac{1}{2}(R_1^{m_{1,i}})^2\sigma_{1,i}^2 \tag{13}$$

Note that the visit of the nth state is always 1. Then we have: $E[X_{1,n}]=1, \text{Var}[X_{1,n}]=0$ and:

$$E[R_n^{x_{1,n}}]=R_n \tag{14}$$

Then the expected reliability following the approach presented in [19] is obtained:

$$E[R]=\left[\prod_{i=1}^{n-1}\left(R_1^{m_{1,i}}+\frac{1}{2}(R_1^{m_{1,i}})^2\sigma_{1,i}^2\right)\right]R_n \tag{15}$$

Effect of the second order of the hierarchy architecture which is to predict the reliability, is obtained by variance summing the visits of service state and reliability information of each service state. Exact prediction of the reliability of a service is done by Taylor series extension. If the effect of the second order is ignored, however, expected reliability of the service can be obtained by the following relation:

$$E[R]\approx\left[\prod_i^{n-1}R_i^{m_{1,i}}\right]*R_n \tag{16}$$

3. EXPERIMENTS AND RESULTS

Figure 1 shows an example of the expected Markov model with the political conditions of the writing operation from the storage service. This service is started from the elementary state n_1 and when the valid input service is received, the writing operation will be done. In this case (state), the failure may occur with regarding to some unexpected failures in the input management (for example overload request and time failure). For transferring the service from the input state n_1 to n_2 , it recalls the policy conditions of a method to know whether the information presented by a user is correctly managed and the input is successfully produced (for simplicity, service model doesn't consider the exceptions). For state n_2 , data, metadata and input are stored in all storage servers. According to the condition, the related policy checks whether the success is returned by all servers. Breach/failure of the policy in this policy result in return to the failure state F . State n_3 shows the final output state from this service for the writing operation. This implicit transmission from n_3 to the final correct step C is occurred with the probability R_3 , subsequently, the operation is fault-free.

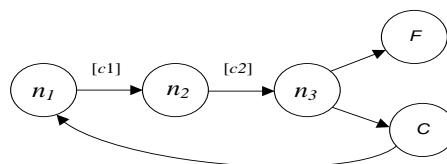


Figure 1. An example of storage service for write operation

If we suppose that the fault-free probability of the nodes calculated by the certification license is: $R_1=0.8154$, $R_2=0.8265$ and $R_3=0.9681$.

And the probability of the transmission among the nodes is $r_{12}=0.5933$ and $P_{r23}=0.7704$, then the matrix corresponding to Q is:

$$Q = \begin{bmatrix} 0 & 0.4838 & 0 \\ 0 & 0 & 0.6367 \\ 0 & 0 & 0 \end{bmatrix}$$

and the matrix M equals:

$$M=(I-Q)^{-1} = \begin{bmatrix} 1 & 0.4838 & 0.3080 \\ 0 & 1 & 0.6367 \\ 0 & 0 & 1 \end{bmatrix}$$

then we have:

$$m_{i,j} = \begin{bmatrix} 1 & 0.4840 & 0.3082 \\ 0 & 1 & 0.6367 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\sigma_{i,j}^2 = \begin{bmatrix} 0 & 0.2342 & 0.0950 \\ 0 & 0 & 0.4054 \\ 0 & 0 & 0 \end{bmatrix}$$

Here we have:

$\prod_{i=1}^{k-1} \left(R_i^{m_{i,i}} + \frac{1}{2} (R_i^{m_{i,i}}) (\log R_i)^2 \sigma_{i,i}^2 \right) = 0.740429$ And $R_k=0.9681$ Probability of the storage service for estimating a

reliability property based on (15) is $E[R] = 0.740429 * 0.9681 = 0.71681$.

Generally, $E[R]$ represents the expected reliability of the service.

Table 1 shows the reliability of each service state and the probability of the transmission among the modeled service states in Figure 1. The number of five defected versions of the program testing step has obtained. For example, defect versions c and d have 1, and the number of two defect state and correct state 3.

Reserving a history of the request of all stored service method in the registry service, reliability of the service and all service states have been estimated and calculated by (17), where f_i is all faults and n_i is the number of the service requests with the service method.

$$R_i = 1 - \lim_{n \rightarrow \infty} \frac{f_i}{n_i} \quad (17)$$

Table 1. Reliability of the service states and transmission probability

Faulty version	Reliability states of service			Transition probabilities	
	1	2	3	P_{12}	P_{23}
A	0.7854	0.8122	0.9168	0.5933	0.7704
B	0.8154	0.8265	0.9681	0.5933	0.7704
C	0.8231	0.8336	1	0.7368	0.6866
D	0.8428	0.8346	1	0.5933	0.7704
E	1	0.8346	1	0.7364	0.6866

Table 2 shows the results obtained from the tests and compares the obtained results with the proposed method and recent work [10, 23]. As seen in Table 2, proposed method can obtain better results than the other methods. Bold values are second best values. It should be noted that the results related to second column of Table 2 have been calculated and the differential fault extent among them based on (18) have been reported.

Figure 2 is the bar graph of the faults of Claudio's and Grass's methods and also the proposed method of this article. In this graph, horizontal axis shows 5 defect versions and vertical axis is the faults of

these versions with different methods. With regarding to this fact that maximum reliability is 1, fault of the program is obtained from (18):

$$\text{Error} = \left| \frac{R_{\text{measured}}}{R_{\text{max}}} - \frac{R_{\text{real}}}{R_{\text{max}}} \right| = \left| \frac{R_{\text{measured}}}{1} - \frac{R_{\text{real}}}{1} \right| = |R_{\text{measured}} - R_{\text{real}}| \quad (18)$$

As seen in Figure 2, proposed method in this paper in the most of cases has fewer faults than Claudio's and Grassi's methods. Moreover the obtained faults have the values adjacent to each other. This enables the stability of the fault values in estimating the reliability of SOA, estimating the probability of the proper operation of the system and predicting the cost of the consequences and losses resulted from non-precision.

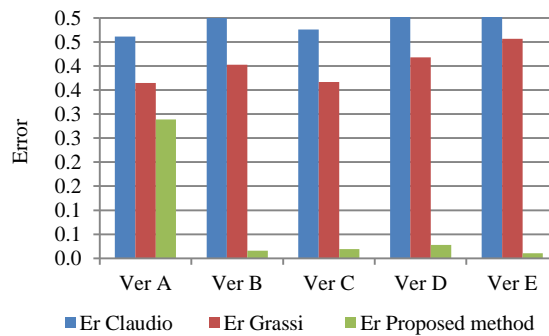


Figure 2. Comparing the errors of proposed method and previous works

Table 2. Reliability prediction comparison

Faulty version	Measured reliability	Claudio's method		Grassi's method		Proposed method	
		reliability	Error	reliability	Error	reliability	Error
A	0.631928	0.17052	0.461408	0.267312402	0.364615598	0.65047	0.288542
B	0.700757	0.20097	0.49978	0.298211514	0.402545486	0.71681	0.016053
C	0.713754	0.23803	0.475724	0.346918841	0.366835159	0.73264	0.018846
D	0.7393	0.22615	0.51315	0.321509293	0.417790707	0.7668	0.0275
E	0.8782	0.35219	0.52601	0.421983976	0.456216024	0.86758	0.0106

4. CONCLUSION

In service-oriented systems, services may be provided, presented and stored by different providers and used by different designers of the operational programs for designing (building) the systems in the various areas of the business process. Reliability of a SOA depends on the reliability of all its individual composed services and input data from the external sources. For studying how to predict the reliability of the services, in this paper we presented a hierarchy method for modeling and analyzing the reliability of the services. Among the major points of this method are high precision of the reliability estimation and also capability of its reuse in using the results of the evaluation of the created products in this step, resulting in saving the time and cost of the evaluation. In this paper, an experiment was explained in which the proposed method has less fault tolerance than the real value compared with the previous method such as Claudio's and Grassi's method. Obtained faults have the adjacent values and this enables the stability of the fault values, estimating the reliability of SOA, estimating the proper operation of the system and predicting the cost of the consequences and the losses resulted from non-precision. When necessary, the services should be tested and simultaneously modified. Of course one of the limitations of this method is that simultaneous modeling the circuit control and implementation of the service in Markov chain model is difficult. Also, this method does not consider the costs in this model. Proper costs should be related to the various services and this method can be used for comparing the cost of the different structures of the system and aid engineers and designers in choosing the optimum system.

REFERENCES

- [1] M. P. Singh and M. N. Huhns, *Service-oriented computing: semantics, processes, agents*: Wiley. com, 2006.
- [2] W.-T. Tsai, "Service-oriented system engineering: a new paradigm," in *Service-Oriented System Engineering*, 2005. SOSE 2005. IEEE International Workshop, pp. 3-6, 2005.
- [3] T. Erl, *Service-oriented architecture*: Prentice Hall Englewood Cliffs, 2004.
- [4] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*: Prentice Hall PTR, 2005.
- [5] M. P. Papazoglou, "Web services and business transactions," *World Wide Web*, vol. 6, pp. 49-91, 2003.
- [6] V. Cortellessa and V. Grassi, "Reliability modeling and analysis of service-oriented architectures," in *Test and Analysis of Web Services*, ed: Springer, pp. 339-362, 2007.
- [7] E. Damiani, S. D .C. di Vimercati, S. Paraboschi, and P. Samarati, "Securing SOAP e-services," *International Journal of Information Security*, vol. 1, pp. 100-115, 2002.
- [8] L. Frantzen and J. T. R. de Vries, "Towards model-based testing of web services," in *International Workshop on Web Services–Modeling and Testing (WS-MaTe 2006)*, 2006.
- [9] L. Wang, X. Bai, L. Zhou, and Y. Chen, "A hierarchical reliability model of service-based software system," in *Computer Software and Applications Conference*, 2009. COMPSAC'0933 .rd Annual IEEE International, pp. 199-208, 2009.
- [10] C. A. Ardagna, E. Damiani, R. Jhawar, and V. Piuri, "A model-based approach to reliability certification of services," in *Digital Ecosystems Technologies (DEST)*, 2012 6th IEEE International Conference on, pp. 1-6, 2012.
- [11] I. Buckley, E. B. Fernandez, M. Anisetti, C. A. Ardagna, M. Sadjadi, and E. Damiani, "Towards pattern-based reliability certification of services," in *On the Move to Meaningful Internet Systems: OTM 2011*, ed: Springer, pp . 560-576, 2011.
- [12] B. Li, Z. Su, Y. Zhou, and X. Gong, "A user-oriented web service reliability model," in *Systems, Man and Cybernetics*, 2008. SMC 2008. IEEE International Conference on, pp. 3612-3617, 2008.
- [13] K. S. Trivedi, *Probability and statistics with reliability, queuing, and computer science applications* vol. 2002: Prentice-hall Englewood Cliffs, 1982.
- [14] G. Delac, M. Silic, and S. Srblic, "Reliability modeling for SOA systems," in *MIPRO, 2012 Proceedings of the 35th International Convention* ,pp. 847-852, 2012.
- [15] M. Rahmani, A. Azadmanesh, and H. Siy, "Architecture-based reliability analysis of web services in multilayer environment," in *Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems* ,pp. 57-60, 2011.
- [16] K. Goševa-Popstojanova and K. S. Trivedi, "Architecture-based approach to reliability assessment of software systems," *Performance Evaluation*, vol. 45, pp. 179-204, 2001.
- [17] R. Billinton and R. N. Allan, *Reliability evaluation of engineering systems: concepts and techniques*: Plenum Press New York, NY, 1983.
- [18] R. C. Cheung, "A user-oriented software reliability model," *Software Engineering, IEEE Transactions on*, pp. 118-125, 1980.
- [19] U. N. Bhat and G. K. Miller, *Elements of applied stochastic processes*: J. Wiley, 1972.
- [20] V. S. Sharma and K. S. Trivedi, "Quantifying software performance, reliability and security: An architecture-based approach," *Journal of Systems and Software*, vol. 80, pp. 493-509, 2007.
- [21] S. S. Gokhale and K. S. Trivedi, "Reliability prediction and sensitivity analysis based on software architecture," in *Software Reliability Engineering, 2002 ISSRE 2003. Proceedings. 13th International Symposium on*, pp. 64-75, 2002.
- [22] V. Grassi and S. Patella, "Reliability prediction for service-oriented computing environments," *Internet Computing, IEEE*, vol. 10, pp. 43-49, 2006.
- [23] V. S. Sharma and P. S. K. Iyer, "Markov Chains and Queueing Networks " Roll No. Y211165, CSE, IIT Kanpur, pp. 1-24, 2002.