Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

SOFTWARE RELIABILITY PREDICTION USING SPN

S. Abbasabadee*[1], H. Motameni [2], H. Jahangirvand[3]

[1]Department of Computer Engineering, Islamic Azad University, Sari Branch, sari, Iran

[2]Department of Computer Engineering, Islamic Azad University, Sari Branch, sari, Iran

[3]Ports and maritime organization, Amirabad port special economic zone, Behshahr, Iran

Published online: 15 May 2016

Abstract

Reliability is an important software quality parameter. In this research for computation of software reliability, component reliability model based on SPN would be proposed. An isomorphic markov chain is obtained from component SPN model. A quantitative reliability prediction method is proposed. The component reliability value is calculated according to the transition cumulative probability distribution of markov chain, obtained from the software SPN model. By means of reliability prediction of the whole software, we'll introduce C_{RM}PN. In C_{RM}PN states are component reliability model and transition are marked with components reliability. With this research more complex software could be simplified and reliability of the software could be evaluated effectively. An example is provided for demonstrating the feasibility and applicability of our method.

Key words: Reliability, SPN, Markov Chain, Component based-software

Author Correspondence, e-mail: abbasabadee@yahoo.com

doi: http://dx.doi.org/10.4314/jfas.v8i3s.149

1. INTRODUCTION

Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment [IEEE, 1983]. With the widespread use of object oriented systems design and development, the use of component–based software development is on the rise. Modern programs can no longer be treated as monolithic entities, but are likely to be made up of thousands or millions of little parts distributed globally, executing whenever

called, and acting as parts of one or more complex systems. Thus predicting the reliability of an application earlier in the life cycle, taking into account the information about Its architecture, and the testing and reliabilities of its components, is essential. In this paper we propose an approach for modeling and quantifying software reliability in software design phase, using stochastic petri net and markov chain concepts. Then we'll present an extended petri net- component reliability model petri net for reliability evaluation of the whole software system. Based on analyzing the petri net structure and relation between component three models are developed and reliability formulas are provided from reduction operation of petri net.

1-1 Related work

Quantitative analysis for software non-functional properties has gained great attention for several decades. Some previous reliability properties analyses are based on markov model [3, 4, 8]. So they are lacking of the ability of express parallelism, synchronizations, confliction and preemption.

To overcome shortcoming of markov chains, petri nets are advised to model and quantifying software nonfunctional parameters [1, 2, 5, 6 , 7,9]

2-Study roadmap

Fig1 shows the roadmap for our study. A component of software is modeled by a SPN. The reachable graph under an initial marking can be gotten. From reachable graph, corresponding makov chain can be extracted. Using the existing technique [8], the cumulative probability distribution can be evaluated for states. Cumulative probability distribution is used for quantitative prediction of software's failure or success probability. We will introduce $C_{RM}PN$ To show the relation between components and make reduction of the net.

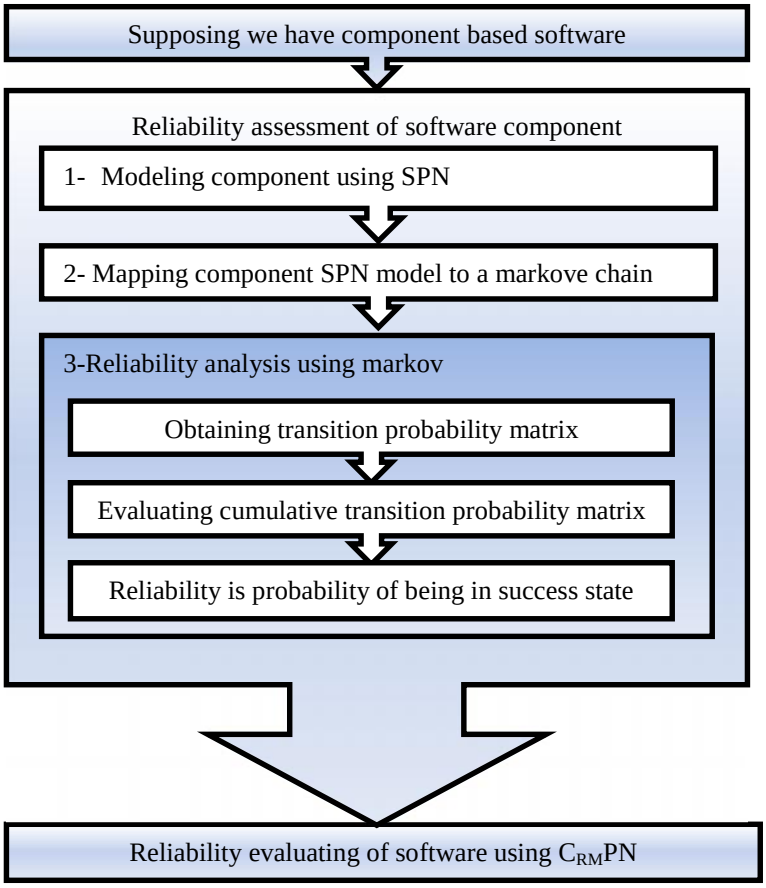


Fig1. Study roadmap

2-1 software component reliability assessment

We will follow these three steps for component reliability assessment

Component reliability modeling using SPN

Suppose that each component may fail. And the failed component can be repaired through some techniques. A software system is composed by several such components. Supposing that a component's failure and repair action probability density functions are negative exponential with rates λ and μ respectively.

Fig.2 demonstrates a reliability evaluation model of a component C_i . The transition t_{fi} represents a failure on the component. The attack rate is λ. Each token appears in the place p_{fi}, named fail place (FP), and denotes that the component C_i has been failed. After being failed, repairing action should be taken, such as rebooting. The transition t_{ri} represents the repairing action after being failed. The rate of the exponential distribution associated with the firing of transition t_{ri} is μ. When the place t_{ei} has a token, it indicates that the component has been executed successfully.

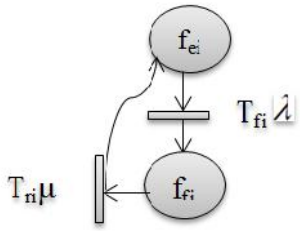


Fig2. SPN component reliability model

Mapping SPN model in to a markov chain

As reachable graph of the SPN component reliability model is isomorphic to markov chain, fig.3 will be obtained.

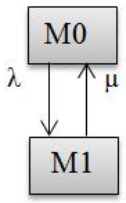


Fig3. Markov chain of the SPN component reliability model

Reliability analysis using markov

By means of reliability prediction using markov chain, we should evaluate the probability of being in success state. So, we follow these steps:

1-computing state transition matrix

Supposing that we have a markove chain isomorphic to the SPN model, and the states number of markov chain is n. Let $Q = [q_{ij}]$, $1 \le i, j \le n$, represents states transition matrix. q_{ij} is defined in Eq. (1). And there exists an arc for i to j

$$q_{ij} = \begin{cases} lable\ value & i \neq j \\ -\sum_{j=1}^{k} lable\ value & \end{cases}$$

There are k arcs depart form i Eq(1)

The state transition matrix of the model can be obtained:

$$Q = \begin{bmatrix} -a & a \\ « & -« \end{bmatrix}$$

2- Evaluating reliability

By means of reliability prediction, Cumulative probability distribution should be gotten from state transition matrix. Eq(2) will be obtained:

$$\begin{cases} P_0'(t) = -\text{a} P_0(t) + \text{«} P_1(t) \\ P_1'(t) = \text{a} P_0(t) - \text{«} P_1(t) \end{cases} \quad \text{Eq(2)}$$

This linear first order differential equation system can be solving by Laplace transform. So, we have Eq(3):

$$f_0 = \frac{s + \text{«}}{s^2 + s\text{«} + s\text{a}} \text{ And } f_1 = \frac{\text{a} f_0}{s + \text{«}}$$

With reverse Laplace transform Eq(4) will be obtained:

$$P_0 = \frac{\text{«}}{\text{«} + \text{a}} + \frac{\text{a}\, e^{-(\text{«} + \text{a})t}}{\text{«} + \text{a}} \text{ And } P_1 = 1 - (\frac{\text{«}}{\text{«} + \text{a}} + \frac{\text{a}\, e^{-(\text{«} + \text{a})t}}{\text{«} + \text{a}})$$

As reliability is probability of being in success state;

R_T= P0

And Eq(5) will be obtained for reliability of a component:

$$R_T = \frac{\text{«}}{\text{«} + \text{a}} + \frac{\text{a}\, e^{-(\text{«} + \text{a})t}}{\text{«} + \text{a}} \quad \text{Eq(5)}$$

Fig4 shows the reliability measure as a function of t, where λ, μ are set to be 0.02, 0.01 respectively. As is shown, the reliability monotonically decreases as time passes. Because the longer the time is, the greater the probability that component fails.

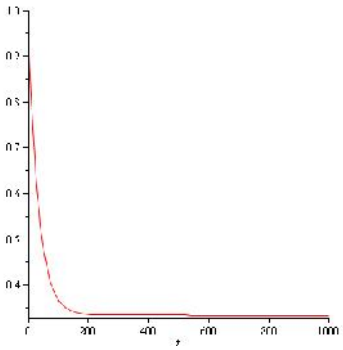


Fig 4. Component reliability measure as a function of t

3- Evaluation of whole software using C_{RM}PN

C_{RM}PN is expanding of general petri net that has component reliability model instead of state and transitions are marked by reliability of each component.

Definition 1: formally, C_{RM}PN can be defined as 4 tuples, C_{RM}PN (C_{RM}, T_{RI}, P, M) where:

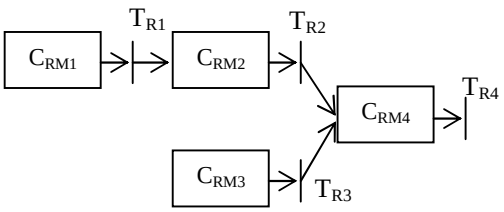
(1) C_{RM} is finite sets of SPN component reliability models, $\forall s : s \in S$

(2) T_{RI} is finite sets of transitions.

(3) $P : T_{RI} \rightarrow R_C$, is the reliability mapping.

(4) $M : P \rightarrow IN\ 0$, is the initial marking.

The flow relation of petri net presents the dynamic action of software [2]. Just because transition own component's reliability in $C_{RM}PN$, software reliability can be evaluate through analyzing the transitions. Fig 5 shows $C_{RM}PN$.



That is equal to:

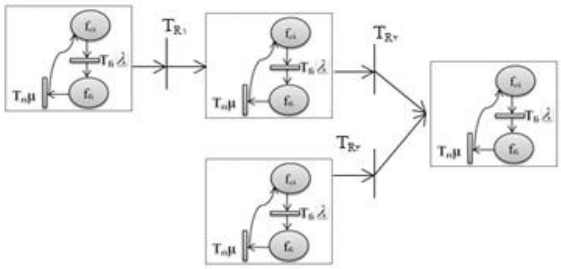


Fig 5. A $C_{RM}PN$

Theorem 1. For any reachable marking M, there is a transaction sequence t1, t2, t3, …tn enables that M_0 [t_1> M_1 [t_2 >M_2 [t_3 >…M[t_n, i.e. $\forall M \in R(M_0)$, $\exists \sigma \in T^*$ $\sigma \rightarrow M_0$ [$\sigma > M$ [11],[12].

From theotrm1, it's learned that when system runs form state M0 to M, software reliability is decided by the transaction sequence σ which it experienced[2].

Software components connection relation

The relationships among components can be directly depicted by T_{RI} s in $C_{RM}PN$ model. There are three typical relations among components in $C_{RM}PN$, i.e. sequential relation, choice relation and merge relation. Each kind of relation can be denoted by an operator, and computation of reliability could be defined.

1. Sequential relation

In sequential model, the whole software reliability relies on each component in the sequence. Any failure component will lead the collapse of whole software.

Definition 1: For a $C_{RM}PN$ composed of n sequential components, the reliability of whole net is the reliability product of all of these sequential components. Then,

$$R_T = R_{C1} \times R_{C2} \times \cdots R_{Cn} = \prod_{i=1}^{n} R_{Ci}$$

where R_{ci} is the reliability of component i.

2. Choice relation

When component are in choice relation, one of these component will be chosen according to some policies. So, reliability for system consisting of components in choice relation could be defined as follows:

Definition 2. For a C_{RM}PN composed of n components in choice relation, the reliability of whole system can be obtained as follows:

$$R_T = R_{C1} \| R_{C2} \| \cdots R_{Cn} = \frac{1}{n} \sum_{i=1}^{n} R_{Ci}$$

where R_{ci} is the reliability of component i.

***3.* Merge relation**

For merged components, the target status could be got by any components. Only when all of these components were failed, would the system be out of operation. Therefore, it's reasonable to define the reliability of the system composed of components in merged relation as following manner:

Definition 3. For a C_{RM}PN composed of n merged components, the reliability of whole net can be obtained as follows:

$$R_T = R_{C1} \oplus R_{C2} \oplus \cdots R_{Cn} = 1 - \prod_{i=1}^{n} (1 - R_{Ci})$$

where R_{ci} is the reliability of component i.

Priority of operator

In a complex software, components relation discussed above usually appear with combined form. So the priority should be defined for these relations. Let R×,R‖,R$\oplus$ respectively represent the priority of sequential relation, choice relation and merge relation, then the follows is held:

R× > R$\oplus$ > R‖

Following this component relation priority rules, one could get the way to calculate the reliability of C_{RM}PN model just like mathematical calculation. Fig 6 shows these relations.

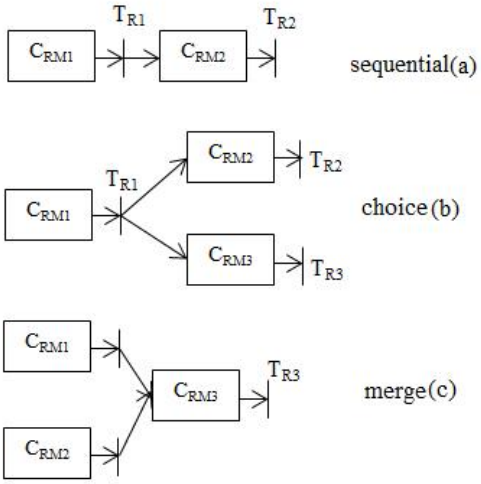
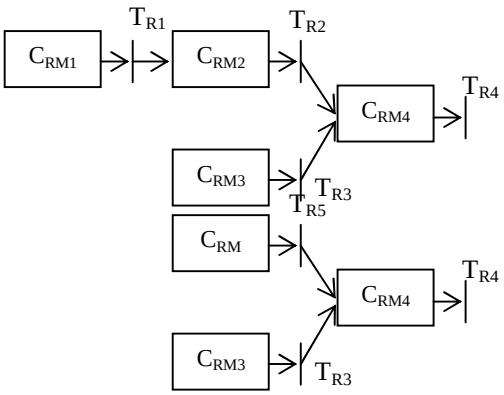


Fig 6. Software component relations in $C_{RM}PN$

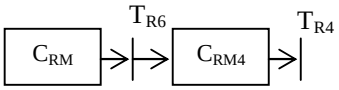
4-Example

Fig 7 shows a $C_{RM}PN$ model for a software system. Model can be simplified by relation rules.

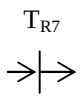
$T_{R5}=T_{R1}\times T_{R2}$



$T_{R6}=T_{R5}\oplus T_{R3}$



$T_{R7}=T_{R6}\times T_{R4}$



$R_T=R_{T7}=(((T_{R1}\times T_{R2})\ \oplus T_{R3})\times T_{R4})$

Table1. Components reliability obtained by Eq(3)

T	T_{R1}	T_{R2}	T_{R3}	T_{R4}
R	$\frac{\alpha_1}{\alpha_1+\beta_1}+\frac{\beta_1 e^{-(\alpha_1+\beta_1)}}{\alpha_1+\beta_1}$	$\frac{\alpha_2}{\alpha_2+\beta_2}+\frac{\beta_2 e^{-(\alpha_2+\beta_2)}}{\alpha_2+\beta_2}$	$\frac{\alpha_3}{\alpha_3+\beta_3}+\frac{\beta_3 e^{-(\alpha_3+\beta_3)}}{\alpha_3+\beta_3}$	$\frac{\alpha_4}{\alpha_4+\beta_4}+\frac{\beta_4 e^{-(\alpha_4+\beta_4)}}{\alpha_4+\beta_4}$

Now, by replacing each components reliability in its corresponding T_{Ri}, reliability of whole software system will be obtained.

5-conclusion

In this paper we present an approach for evaluating whole software system based on SPN and markov chain concepts.

5. REFRENCES

1- Nianhua Yang, Huiqun Yu, Zhilin Qian, Hua Sun " Modeling and quantitatively predicting software security based on stochastic Petri nets "Advanced Theory and Practice for Cryptography and Future Security Volume 55, Issues 1–2 (January 2012) Pages 102-112

2-YU Ruiqiang, HUANG Zhiqiu "Operators for analyzing software reliability with petri net" international symposium of information science and engineering 970-0-7695-3494-7/08 DOI10.1109/ISISE20008.100 2008 IEEE

3-Gyula Zsigmond, Szilvia Homolya, Marianna Lendvay "application of continuous time markov chain by reliability analysis"978-1-4244-5349-8/09/2009IEEE

4-Ravil I.Muhamedyev" simulation models of reliability estimation of information systems" proceeding of the 9th international conference "reliability and statistics in transportation and communication"(relstat'9), 21-24 October 2009, p424-428. ISBN978-9984-818-21-4

5-NSato,K.S. Trivedi "stochastic modeling of composite web services for closed form analysis of their performance and reliability bottlenecks" ICSOC 2007,LNCS 4749,PP107-118,2007 springer-verlag berlin Heidelberg 2007

6-Tingjun Thi, Jingfeng Zhao, Xinchun Yin, Jiajia Wang" research on telecommunication switching system survivability based on stochastic petri net" the 3th international conference on innovative computing information and control(ICICIC'08)978-7695-3161-8/08 /2008IEEE

7-Xiaojing Hu, Shixi Liu, Lisheng Ma" research on dependability of virtual computing system based on stochastic petri nets"978-1-4244-7237-6/10/2010 IEEE

8-Hoang Pham" system software reliability" Springer London Ltd, ISBN1852339500 published In November 2 2006

9-H.motameni, A.movaghar, M.fadavi Amiri"Mapping Activity diagram to petri net: application of markov theory for analyzing nonfunctional parameters" IJE Transaction vol.20, no.1, April 2007-65

How to cite this article:
Abbasabadee S, Motameni H, Jahangirvand H. Software reliability prediction using spn. J. Fundam. Appl. Sci., 2016, *8(3S), 956-965.*