Journal of Advanced Computer Science & Technology, 3 (2) (2014) 221-231
©Science Publishing Corporation
www.sciencepubco.com/index.php/JACST
doi: 10.14419/jacst.v3i2.3663
Research Paper

Reliability evaluation of a payment model in mobile e-commerce using colored Petri net

Abdolghader Pourali [1]*, Mohammad V. Malakooti [1], Muhammad Hussein Yektaie [2]

[1] *Islamic Azad University, UAE branch, Dubai, UAE*
[2] *Islamic Azad University, Abadan branch, Abadan, Iran*
**Corresponding author E-mail:*

Copyright © 2014 Abdolghader Pourali et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In general, a mobile payment system in mobile e-commerce must have high security so that the user can trust it for doing business deals, sales, paying financial transactions, etc. in the mobile payment system. Since an architecture or payment model in e-commerce only shows the way of interaction and collaboration among users and mortgagers and does not present any evaluation of effectiveness and confidence about financial transactions to stakeholders, in this paper we try to present a detailed assessment of the reliability of a mobile payment model in the mobile e-commerce using formal models and colored Petri nets. Finally, we demonstrate that the reliability of this system has high value (case study: a secure payment model in mobile commerce presented at the International Conference) [18].

Keywords: *Reliability, Colored Petri Net, Assessment, Payment Models, M-Commerce.*

1. Introduction

Developing a model for software systems before its build or rebuild is as important as having a blueprint for a building. Many engineering disciplines draw how products should be built and take great care that their products are manufactured according to these models and descriptions. Good and accurate models play an important role in establishing complete relationships between people in a project. Perhaps complex systems are modeled because they cannot be visualized all at once, thus they are model led so that they are fully understood and the relationships between their various parts are displayed. The Unified Modeling Language (UML) is a language for modeling or producing software development maps. In this study, we first implement the proposed model using UML. Since this model is a quasi-formal model, it is not possible to directly assess its efficiency and reliability. Therefore, it is required to convert the quasi-formal model to a formal one. To assess the reliability and develop a model, colored Petri nets are used due to their ease of use, high performance and taking advantage of the strong math background. Finally, using CPNTools, the resulted Petri net is simulated and the evaluation results are used for reliability metrics.

2. Work context

The presentation of the proposed model based on UML, colored Petri networks, simulation using CPNTools.

2.1. Unified modeling language (UML)

UML is an international language which uses graphic symbols to display different aspects of a system's performance in a way that is easy to understand for all people involved in the system. UML is a language for specifying, visualizing, constructing and documenting software and non-software systems as well as for modeling business systems [1].
Using diagrams, UML shows the structure and behavior of systems. A system is modeled as a set of discrete objects, interacting to achieve the ultimate goal of the system. The UML diagrams display two different views of the system [2].

- Static (structural) view: This view focuses on displaying the static structure of the system using objects, features, operations and relations. Structural diagrams display different types of objects that are important for the system and its implementation, and the relationships between objects such as the "class diagram."
- Dynamic (behavior) view: This view focuses on displaying the dynamic behavior of the system through showing the coordination among objects and changes to the internal state of objects such as "sequence diagram" or "activity diagram."UML 2.2 has fourteen varieties of diagrams divided into two main categories: seven diagrams to represent structural information and seven others to display the behavior and interactions.

Structural diagrams: Structural diagrams focus on what must be included in the desired system. Since these diagrams show the structure, they are widely used in the architecture of software systems, such as Class Diagram, Component Diagram, Composite Structure Diagram, Deployment Diagram, Object Diagram, Package Diagram, and Profile Diagram.

Behavior diagrams: Behavior diagrams focus on what must happen in the modeled system. Since these diagrams represent the system's behavior, they are widely used to describe the functions of software systems, such as Activity Diagram, UML State Machine Diagram, and Use Case Diagram.

2.2. Petri net

Petri nets [3] were introduced by Carl Adam Petri (1962). It is a formal visual method for the description and modeling of dynamic, controlled, distributed, and concurrent systems and communication systems. Petri nets and their extensions are widely used for modeling, description and architecture of systems [4,5,6] and safety analysis of systems[7.17] Using Petri nets based on the graph theory, one can also analyze systems with asynchronous, non- deterministic and stochastic structure. The Petri net is based on a graph. Informally we can say that it is a bipartite directed graph which consists of two elements: location and transition. A formal description of the Petri net is as follows.

Definition 1: Petri nets an arranged five in the form of (P, T, I -, I +, M0), where

1) P is a finite set of locations
2) T is a finite set of transitions
3) P ∏ T=Ø
4) P*T→N, +I and -I are called forward and backward functions.
5) P→N0: M0 is called the initial transition function.

Functions I+ and I- represents the relationship between locations and transitions. If I-(P, T)>0 is satisfied, then there is an arc from location P to transition T. In this case, location P is called input location. If I+ (P, T) > 0 is satisfied, then there is an arc from transition to T location P. In this case, P is called output.

Intersection functions assign natural numbers to locations. These numbers are called arc weights. If each input from transition Contains nuts at least equal to the number of arc weights that connect it to transition T, it is said that transition T is activated.

3. Related work

Today many efforts have been done by researchers in the context of converting quasi-formal languages into formal methods such as types of Petri nets. In [8], the Markov theory is used for analysis of non-operational parameters of a software system. To analyze the stochastic behavior of the system, first the UML diagram is mapped to the generalized stochastic Petri net and then the efficiency features such as security and reliability are evaluated by the generalized stochastic Petri net. In [9], an algorithm for mapping the sequence diagram to the fuzzy Petri net is proposed which allows for performing the verification and validation operations of the system's quality parameters with greater precision. Using the proposed method, the study assessed the quality parameters of reliability. In [10], an algorithm is developed to map the sequence diagram of fuzzy unified modeling language to the Petri net. This formalization adds the automated processing ability to the quasi-formal unified modeling language and sets the time required for washing in case of uncertain events, during a case study for washing machines. In [11], the sequence diagram is converted to a high value fuzzy network. Thus HLPN is used to show the formal semantic framework for unified modeling language and combine its structural and behavioral characteristics but the parameters have not been measured. In [12], the activity diagram is converted to the fuzzy Petri net but parameters have not been measured. In [13], the relationship diagram is converted to the fuzzy Petri net but the parameters have not been measured. In [14], the activity diagram is converted into the fuzzy Petri net, and reliability assessment is used. The state diagram in UML represents a situation that may occur for an object. In [15], for modeling the dynamic behavior of the system, the UML collaboration diagram is mapped to the fuzzy Petri net using an algorithm but parameters are not measured. In [16], to evaluate the system throughput, an algorithm is proposed to map the sequence diagram to the fuzzy Petri net.

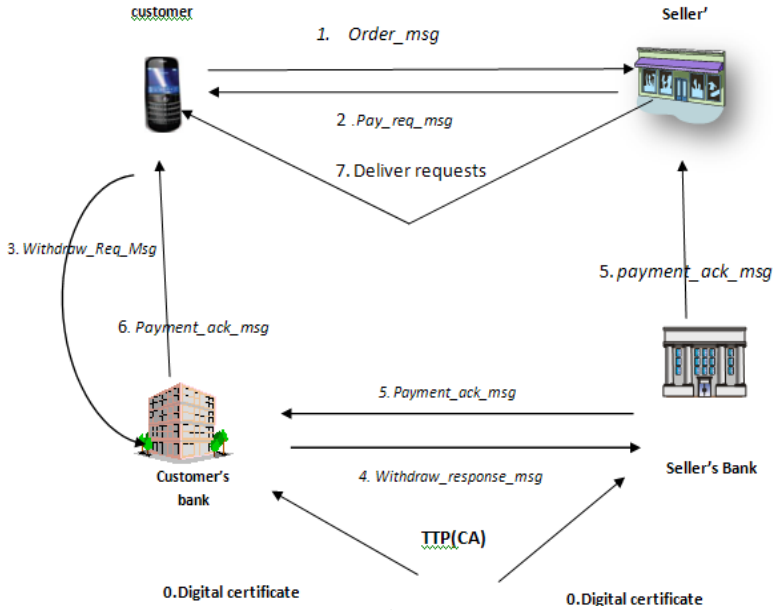


Fig. 1: Proposed Model for Reliability Evaluation

4. Evaluation of reliability

Case Study: Figure (1) shows a secure model in mobile e-commerce [18] in which important metrics such as data integrity, Non-Repudiation, authentication and Confidentiality are satisfied. Now, we want to provide an accurate assessment of the system reliability before and after encryption using UML, Petri net and CPNTools. The procedure is as follows.

4.1. Calculating the reliability of model 1 before applying the encryption on messages:

Step 1: Obtaining the sequence diagram of Figure 1:

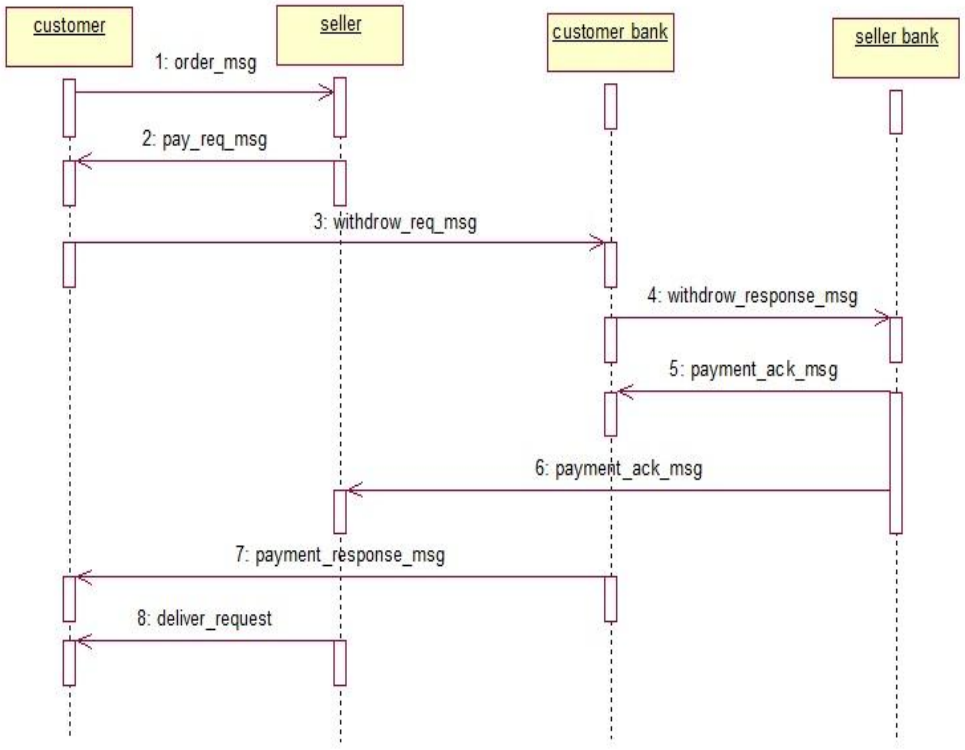


Fig. 2: Sequence Diagram Resulted from (Fig. 1)

Step 2: We calculate the reliability of each exchanged message in the above sequence diagram (Figure3).

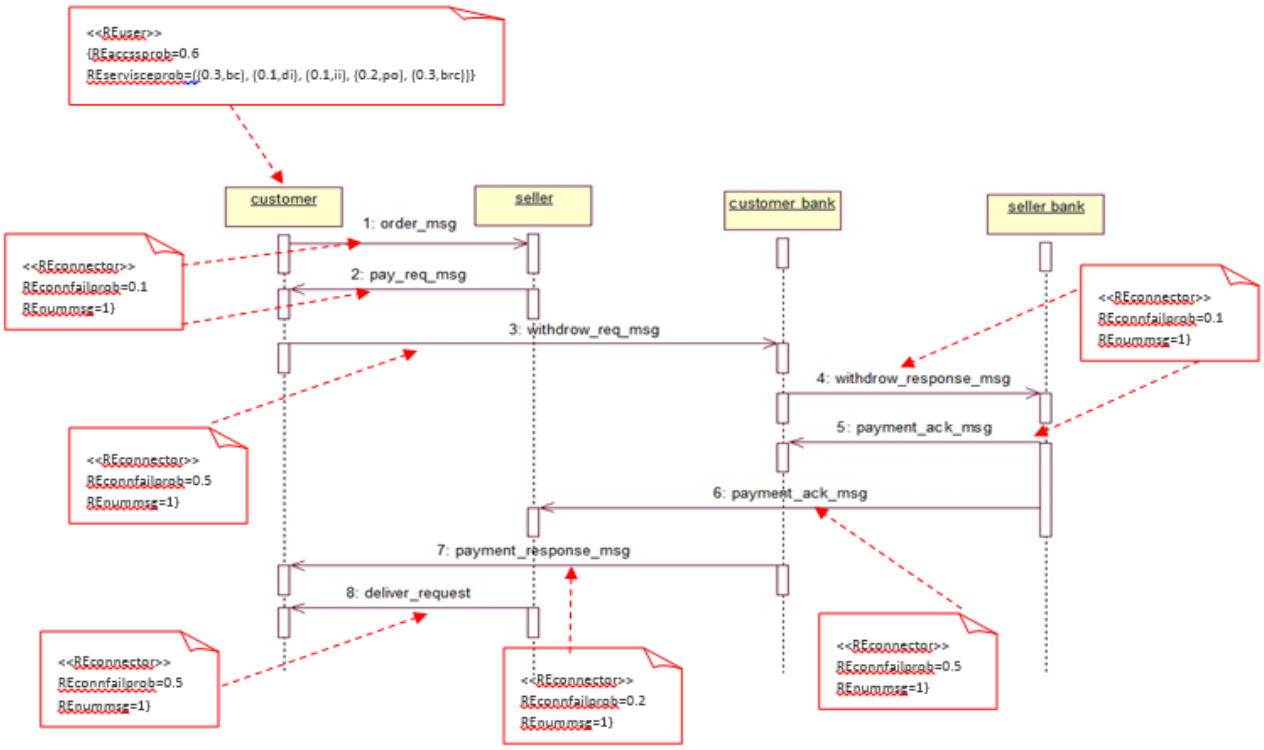


Fig. 3: Calculation of the Reliability of Sent Messages

Step 3: Calculation of the reliability between components:

- Probability of failure = p
- Probability of success = 1-p
- Overall reliability = 1-$\sum$ pi

Message No.	Failure Rate	Success Rate
1	0.1	0.9
2	0.1	0.9
3	0.5	0.5
4	0.1	0.9
5	0.1	0.9
6	0.5	0.5
7	0.2	0.8
8	0.5	0.5
	26%	73%

To cover this low confidence percentage (73%) and satisfy all four metrics, the proposed method is used which is based on encryption. It will be presented in the next section.

Step 5: Petri net equivalent to the sequence diagram.

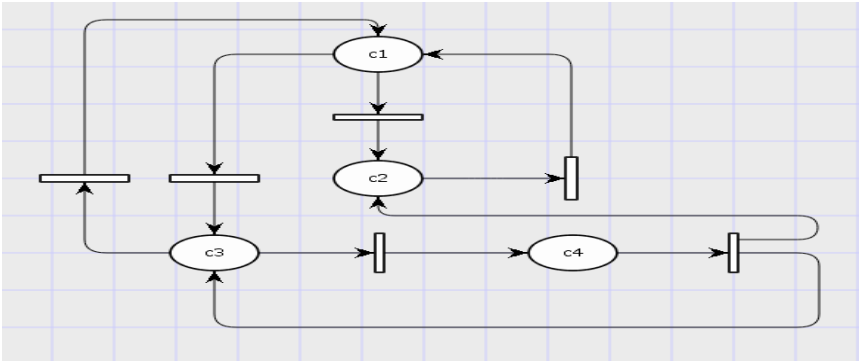


Fig. 4: Petri Net Resulted from the Sequence Diagram (Fig. 2)

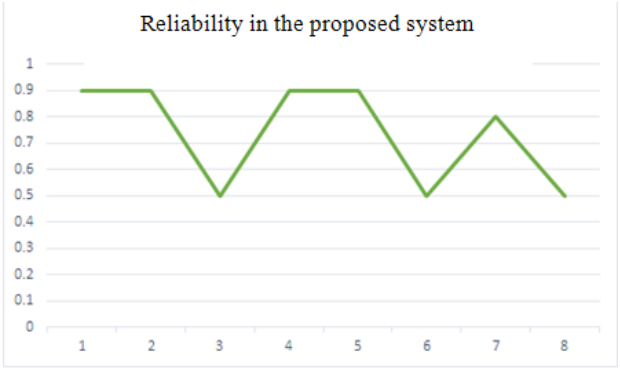


Diagram 1: Calculation of Reliability in the Proposed Model after Encryption

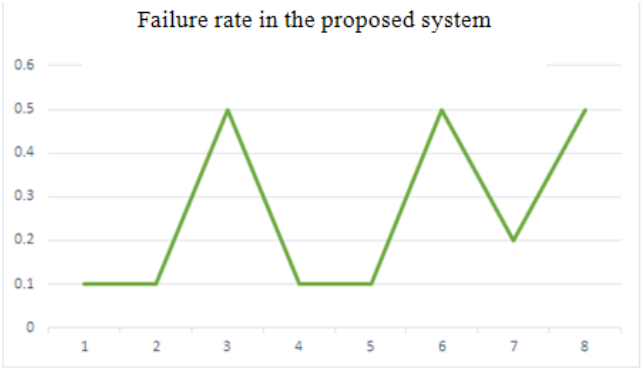


Diagram 2: Calculation of Failure Rate in the Proposed Model before Encryption

Step 6: Petri net implementation using CPN Tools

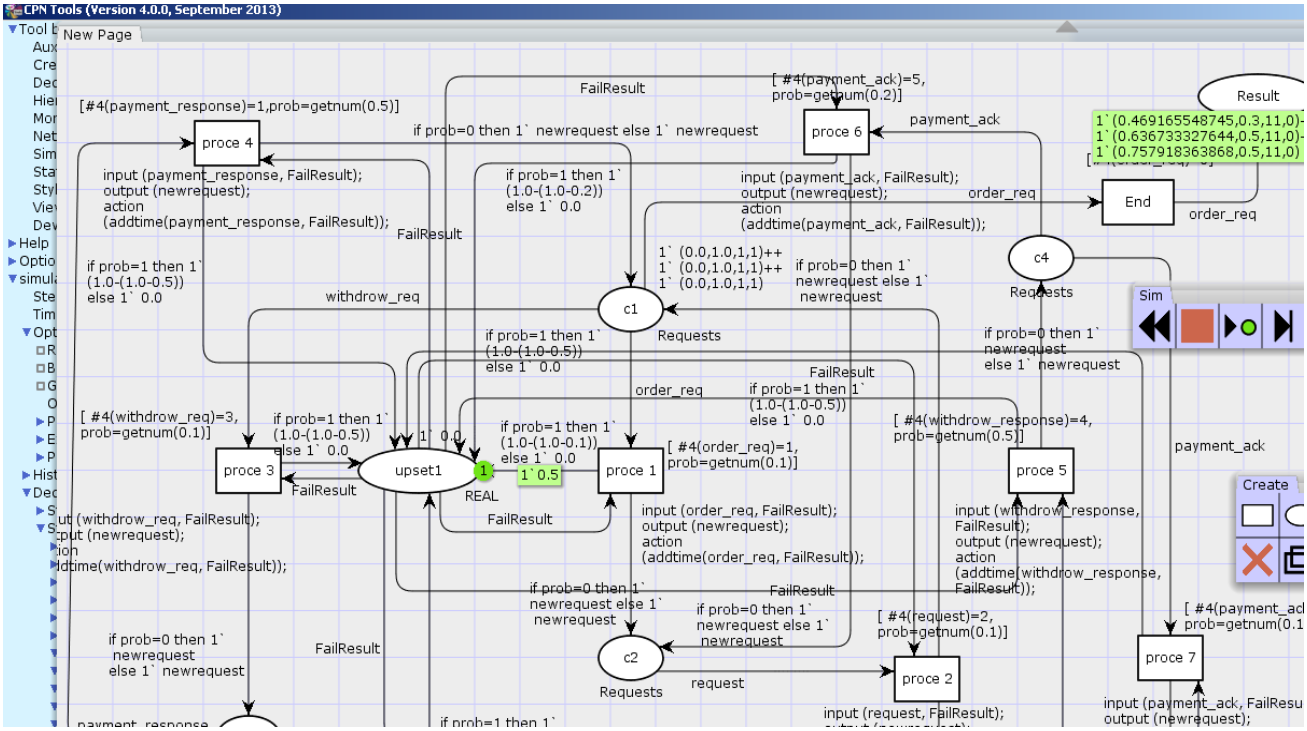


Fig. 5: Calculation of Reliability in the Proposed Model Using Cpntools as Seen in Figure 5, the Overall System Reliability for 3 Users Is Less than or Equal To 50%. To Cover this Low Probability, We Recalculate the Reliability of the Proposed Method by Applying Encryption Algorithms on All Messages and Evaluating Important Metrics Such As Data Integrity, Denial, Authentication and Privacy. The Procedure Is as Follows.

4.2. Calculation of the model reliability (Figure 1) after applying encryption on messages

For evaluating the reliability of the system that can meet the needs of stakeholders, the proposed method uses formal models and presents an applicable model using colored Petri nets. In the desired architecture, we use collaboration diagrams that show the architecture behavior and the component diagram that shows the architecture structure. Then we draw the colored Petri net model and annotate the reliability evaluation capabilities of the system using annotation methods. To calculate the system reliability, we consider failure among components. In other words, total failure represents the system reliability. Finally, using the resulted Petri nets, we simulate and calculate reliability with CPN Tools.

Step 1: Component diagram resulted from Fig 1

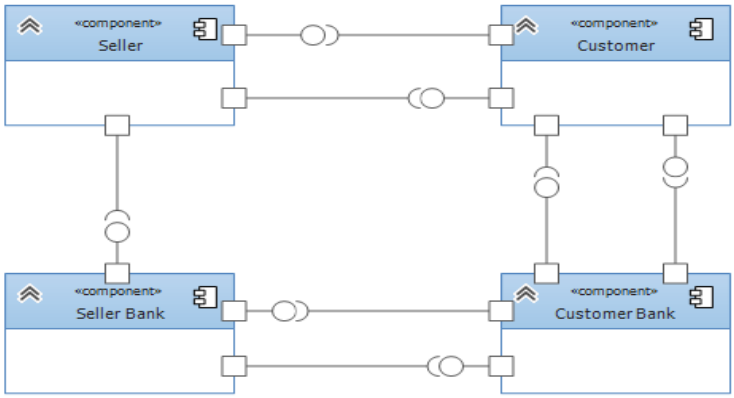


Fig. 6: Components Diagram Resulted from Fig 1

Step 2: Collaboration diagram resulted from Fig. 1

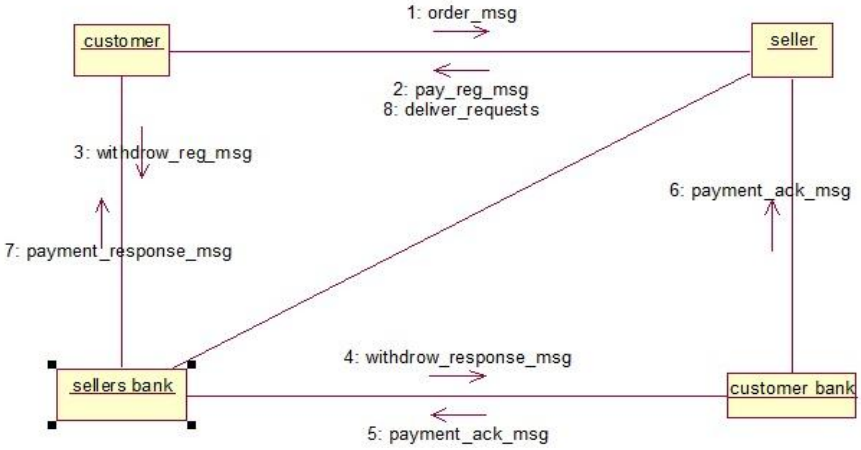


Fig. 7: Collaboration Diagram Resulted from Fig 1

Step 3: Annotation using reliability stereotypes

<<REcomponent>>Stereotype: Table (1) shows the base class and tags of this stereotype. The REcompfailprob tag shows merely the failure probability of each component and REbp shows the number of calls of a component (busy period). For nested components, the REcompfailprob tag of external components is a function of REcompfailprob for more internal components.

Table 1: The <<Recomponent>>Stereotype and Its Tags

Stereotype	Base Class	Tags
<<REcomponent>>	Classifier ClassifierRole Component Instance	REcompfailprob REbp

Tag	Type	Multiplicity
REcompfailprob	Real (0,1)	[0..1]
REbp	Integer	[0..*]

The <<REconnector>> stereotype: Table (2) shows the base class and tags for this stereotype. The <<REconnfaiprob>>tag shows the failure probability of a connection, and REnummsg shows the number of calls of an interface (i.e. the number of messages).

Table 2 : << Reconnector>> Stereotype and Its Tags

Stereotype	Base Class	Tags
<<REconector>>	Message Stimulus associationRole	REconnfailprob REnummsg

Tag	Type	Multiplicity
REconnfailprob	Real (0,1)	[0..*]
REnummsg	Integer	[0..*]

We obtain the annotated component diagram with reliability stereotypes. The failure probability of each component is calculated 0.1 with UML data in the site of evaluation of non-functional requirements [19] and article [20],[21].

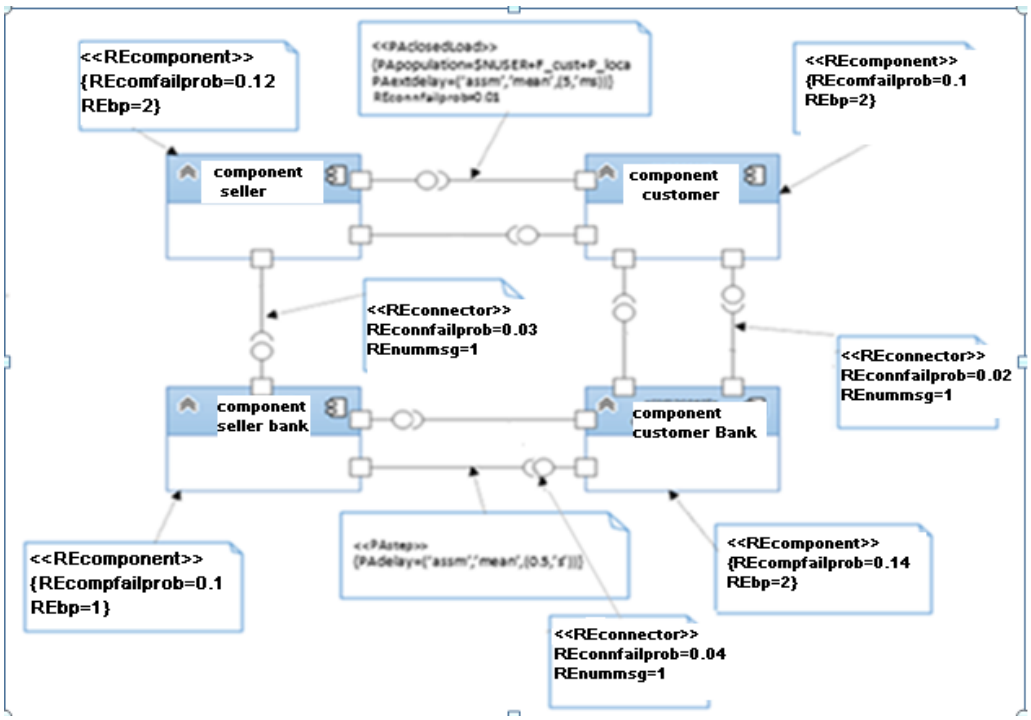


Fig. 8: Annotated Component Diagram Resulted From Fig. 6

Table 3: Summarizes the Data Obtained From the Diagram in Fig. 8

Component Code	Component Name	θ_i	$bp_{i,j}$	linteract(l, i, j)
C1	Customer	0.10	2	4
C2	Seller	0.12	2	3
C3	Customer bank	0.14	2	4
C4	Seller bank	0.10	1	3

Step 4: Obtained colored Petri net

Assuming that customer component is represented by C1, Seller component by C2, Customer bank by C3 and Seller bank by C4, the obtained Petri net is as follows.

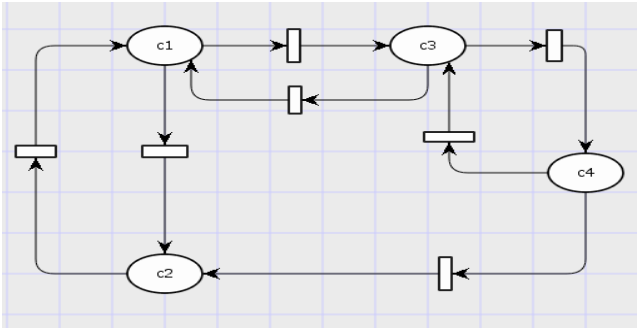


Fig. 8: Petri Net Diagram Resulted From Fig. 1

Step 5: How to calculate reliability

In calculating the failure probability of a component, the number of busy periods that the component c_i has in diagram j is enumerative and is shown by the variable bp_{ij}. The failure probability of component c_i in diagram j is represented by Θ_{ij} and is calculated according to the following equation:

$$\theta_{i,j} = \text{prob(failure of Cij)} = 1 - (1 - \theta_i)^{\text{bpij}} \quad (1)$$

To calculate the failure probability between the two components m and 1, if nms(1, m, j), then the number of interactions between these two components in diagram j is represented by linteract(1, m, j). Thus, using the failure probability between the two components (ψ_i), the reliability ψ_{mj} is calculate

$$\psi_{1mj} = (1 - \psi i)^{\text{linteract(1,m,j)}} \quad (2)$$

Once the failure of each component or between two components was calculated for each transition, we use the following equation to calculate the average failure probability in each transition. It calculates the reliability of the system:

$$\theta_s = 1 - \sum_{j=1}^{k} pj \left[\prod_{i=1}^{N} (1 - \theta_i)^{\text{bpij}} \Pi \left(1 - \psi_{lij}\right)^{\text{linteract(i,j,k)}}\right] \quad (3)$$

Finally, all transitions are calculated with the same failure probability and service time. To calculate the overall system reliability, we calculate total failures and service times in all transitions. The parameters of Equation (3) are summarized in Table 4.

Table 4: Definition of the Parameters Resulted From Equation 3.

θ_i	is the failure probability on demand of a software component i
bpij	number of busy periods in component c_i
linteract (l, i, j)	number of interactions between two components
ψl	failure probability between two components
θ_s	is the failure probability on demand of the whole system
N	is the number of software components
K	K is the number of system scenarios;

Example: According to equation 1 and the data in Table 3, the Θ_{ij} values for Customer, Seller, Customer bank, and Seller bank are computed as follows.

$\theta_{i,j} = \text{prob(failure of Cij)} = 1 - (1 - \theta_1)^{\text{bpij}}$

= 1 -$(1 - 0.1)^2$

=0.19,

$\theta_{i,j} = \text{prob(failure of Cij)} = 1 - (1 - \theta_2)^{\text{bpij}}$

= 1 -$(1 - 0.12)^2$

=0.22,

$\theta_{i,j} = \text{prob(failure of Cij)} = 1 - (1 - \theta_3)^{\text{bpij}}$

= 1 -$(1 - 0.14)^2$

=0.26,

$\theta_{i,j} = \text{prob(failure of Cij)} = 1 - (1 - \theta_2)^{\text{bpij}}$

= 1 -$(1 - 0.10)^1$

=0.1

According to Equation 2 for the components Seller and Customer, reliability is calculated as follows:

$\psi_{1mj} = (1 - \psi i)^{\text{linteract(1,m,j)}}$

$(1 - 0.01)^4$ =

0.96 =

$\psi_{1mj} = (1 - \psi i)^{\text{linteract(1,m,j)}}$

$(1 - 0.03)^3$ =

= 0.91

Reliability in the proposed system

Parameter	Components Name	Failure Rate	Reliability
$\psi 1$	Customer	0.01	0.96
θ_{ij}	Customer	0.19	0.81
$\psi 2$	seller	0.03	0.91
θ_{ij}	seller	0.22	0.78
$\Psi 3$	Customer bank	0.02	0.92
θ_{ij}	Customer bank	0.26	0.74
$\Psi 4$	Seller bank	0.04	0.88
θ_{ij}	Seller bank	0.1	0.99

Average total reliability of the system

Average Failure Rate	0.10%
Average Reliability	0.87 %

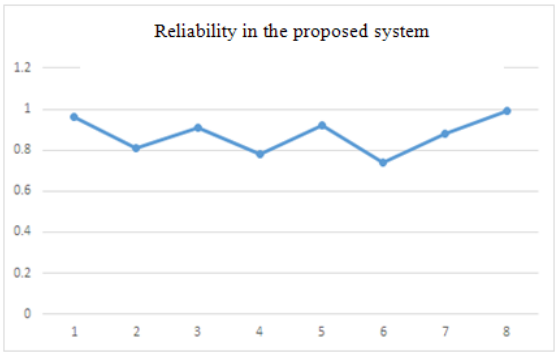


Diagram 3: Calculation of Reliability in the Proposed Model after Encryption

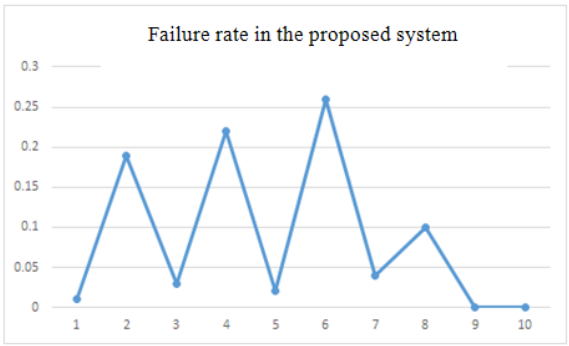
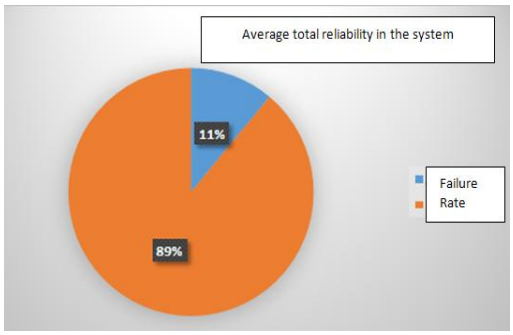


Diagram 4: Calculation of Failure Rate in the Proposed Model after Encryption



Step 6: Simulation

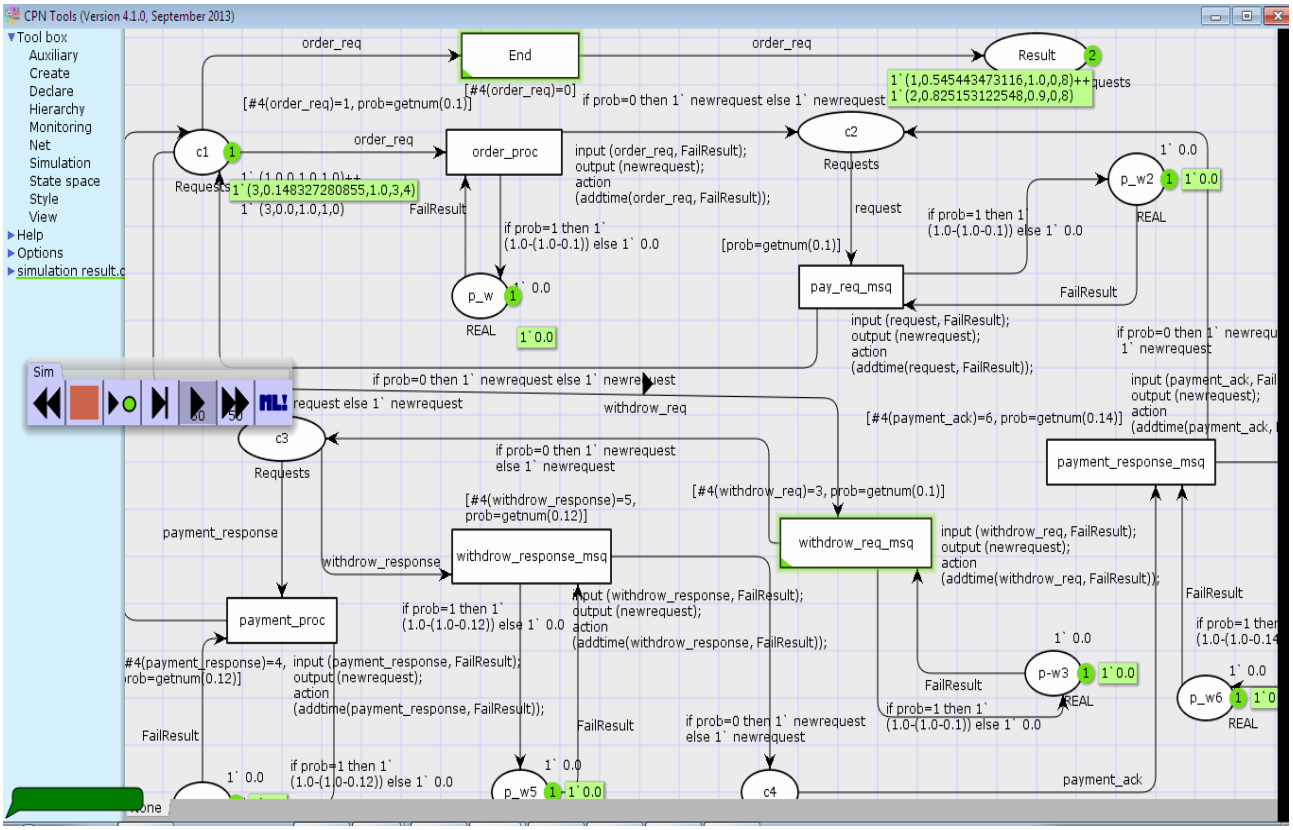


Fig. 9: Petri Net Simulation Using Cpntools

5. Conclusion

This study provides the reliability assessment for the proposed model or any other architecture by mapping the system to the Petri net. To achieve this, the studied system is first converted to UML diagrams and then to a Petri net with the help of simulation and CPNTools, we obtain a precise evaluation of non-functional requirements such as performance. Therefore, according to the results of simulation and mathematical formulas, the proposed system with its security parameters is considered a reliable model.

References

[1] http://www.omg.org/gettingstarted/what_is_uml.htm.

[2] J. Rambaugh, G. Booch, I. Jacobson; "The Unified Modeling Language Reference Manual 2nd Edition" Addison-Wesely, 2005, ISBN 0-321-24562-8

[3] David, René, Alla, Hassane, "Discrete, Continuous, and Hybrid Petri Nets", ebook, Springer, 2010. http://dx.doi.org/10.1007/978-3-642-10669-9.

[4] A. Bobbio, M. Gribaudo, A. Horv´ath, "Modeling a car safety controller in road tunnels using hybrid petri nets", international IEEE conference on Intelligent Transportation Systems, ITSC '06, IEEE.

[5] K. Fukuzawa and M. Saeki, "Evaluating software architectures by cloud red Petri nets ",in Proceeding SEKE '02 Proceedings of the 14th international conference on Software engineering and knowledge engineering, Pages 263-270.

[6] Z. S. Dehkordi, A. Harounabadi, S. Parsa, "Evaluation of software architecture using fuzzy color Petri net", Growing Science TM publisher of academic and scientific journals, Volume 3 Issue 2 pp. 555-562, 2013.

[7] G. Ramos, J. L. Sánchez, A. Torres, M. A. Ríos. "Power Systems Security Evaluation Using Petri Nets", Power Delivery, IEEE Transactions on Volume: 25, Issue: 1, pp 316 - 322, 2010. http://dx.doi.org/10.1109/TPWRD.2009.2035422.

[8] H. Motameni, A. Movaghar, M. Fadavi Amiri, "Mapping activity diagram to Petri net application of Markov Theory for analyzing non-functional" parameters, International Journal of Engineering Transaction B: Applications, vol. 20, pp. 65-76, 2007.

[9] H. Nematzadeh, S.Deris, H. Maleki, Z. Nematzadeh, "Evaluating Reliability of System Sequence Diagram Using Fuzzy Petri Net", International Journal of Recent Trends in Engineering, Issue. 1, Vol. 1, May, pp. 142-147, 2009.

[10] R. Noorian Talouki and 2H.Motameni. "Modeling Sequence Diagram in Fuzzy Uml to Fuzzy Petri-Net for Calculating Reliability Parameter". Research Journal of Applied Sciences, Engineering and Technology 6(20): 3703-3714, 2013.

[11] A.Alhroob, K.Dahal, A.Hossain, "Transforming UML Sequence Diagram to High Level Petri Net", 2nd International Conference on Software Technology and Engineering (ICSTE), PP 260-264 2010.

[12] H. Motameni, A. Movaghar, I. Daneshfar, H. Nemat Zadeh, "Mapping to Convert Activity Diagram in Fuzzy UML to Fuzzy Petri Net", World Applied Sciences Journal 3 (3): 514-521, 2008.

[13] H. motameni T. Ghassempouri, "Transforming. Fuzzy Communication Diagram to Fuzzy Petri Net", ISSN 1450-223X Issue 16, pp.62-73, 2011.

[14] H. Nematzadeh, Deris, Safaai, "Reliability testing of UML activity diagram using Petri net", Proceedings of Intenational Conference on Computer Science and Applied Mathematics (ICCSAM), Italy, 2009.

[15] H. Nematzadeh, T. Malaysia, M .Fallah, H. Motameni, "Transforming FUZZY Collaborated Diagram to Fuzzy Petri Net", American Journal of Scientific Research ISSN 1450-223X Issue 19, pp. 45-50, 2011.
[16] H.Motameni, H.Nematzadeh, E.Mesgarian, "Evaluating Throughput of System Sequence Diagram Using Fuzzy Petri Net", International Conference on Recent Trends in Computer and Information Engineering (ICRTCIE') April 13-15, Pattaya, 2012.
[17] F. TIAN, J. YAN, "Study of Mobile E-Commerce Safety System Model Based on Petri Net", Journal of Convergence Information Technology (JCIT) Volume8, Number10, May 2013.
[18] A. Pourali, Dr. M. V. Malakooti, Dr. M.H Yektai, " A Secure SMS Model in E-Commerce Payment using Combined AES and ECC Encryption Algorithms", Proceedings of the International conference on Computing Technology and Information Management, Dubai, UAE, pp 431-442,2014.
[19] http://www.omg.org/.
[20] S. Bernardi, J. Merseguer, "Performance evaluation of UML design with Stochastic Well-formed Nets", The Journal of Systems and Software 80 (2007), sciencedirect, PP 1843–1865.
[21] S. Emadi1, F. Shams," Modeling of component diagrams using Petri nets", Indian Journal of Science and Technology Vol. 3 No. 12 (Dec 2010),