Management Science Letters 2 (2012) 2417–2424

Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

A method for enterprise architecture validation with colored Petri Nets

Mohammad Sadegh Alishahi[a*], Ali Harounabadi[b] and Seyed Javad Mirabedini[b]

[a]*Department of Computer, Science and Research Branch Bushehr, Islamic Azad University, Bushehr, Iran*
[b]*Department of Computer, Central Tehran Branch, Islamic Azad University, Tehran, Iran*

ARTICLEINFO

Article history:
Received March 22, 2012
Received in Revised form
June, 15, 2012
Accepted 28 July 2012
Available online
July 30 2012

Keywords:
Enterprise Architecture
Architectural Validation
C4ISR Framework
Colored Petri Nets

ABSTRACT

Architecture includes so many documents where each describes one part of an enterprise. The problem in using such descriptions is on how to consider and use all components. Therefore, in order to organize the descriptions of enterprise architecture, we should use a framework. C4ISR is one of the enterprise architectural frameworks, which includes three views, contains some products. In order to show the products, this framework needs a unified notation, which covers all the products with various views. Unified Modeling Language (UML) prepares such situation. But in order to decrease the expenses of enterprise architectural productions process, the architectural products shall be evaluated before the architectural implementation level happens. In this article, a simple way for validation of enterprise architectural products with Colored Petri Nets is presented to evaluate true behavior of architectural products well.

1. Introduction

The necessity of enterprise architecture is normally evaluated based on the appearance of great enterprises and architecture needs to design and develop complex information systems. It needs to design appearance of information systems with special goals and the importance of enterprises flexibility against outer pressures including business change, enterprise structures and missions change and fast technology changes. Architecture includes so many documents that each of them describes one part of the enterprise. The problem happens in using such descriptions is to understand how we could consider and use all components, properly. Therefore, in order to organize the descriptions of enterprise architecture, we should use a framework. Although in the most texts, Zachman Framework is mentioned as the first architectural framework of information site, it shall be said that the attempts of American Defense Ministry for communication and information sites are separate from the Zachman Framework and its issues.

* Corresponding author.
E-mail addresses: m_s_alishahi@yahoo.com (M. Sadegh Alishahi)

doi: 10.5267/j.msl.2012.07.026

Contrary to Zachman Framework, which includes six views, C4ISR framework includes three views, which are totally different from each other. The concept of these three views is different from what is known as view in Zachman framework. The reason for this difference is that, Zachman framework does not consider the process of work performance while C4ISR framework is mostly based on work performance process (Khayami, 2011).

During the last years many attempts were done in the field of evaluation and validation of enterprise architecture (Locob & Jonkers, 2005; Wagenhals et al., 2003; Rezaei & Shams, 2008; Ostadzadeh, & Nekoui, 2009; Khayami et al., 2011). Locob and Jonkers (2005) presented an approach for layered quantitative analysis and demonstrated enterprise architectural models based on service, which includes two phases: an up-down propagation of workload and a down-up propagation of expense measurement or implementation.

Wagenhals et al. (2003) developed a general description out of architectural process based on object-orientation and UML and the principles of using UML products to describe architecture within C4ISR framework. They also developed a map among UML products and an executive model made based on Colored Petri Nets for evaluating architectural executive, behavioral and logic evaluation. In the present research, the method of covering C4ISR products with UML diagrams is investigated, which explains how the UML diagrams are mapped to Colored Petri Nets and they are evaluated.

Rezaei and Shams (2008) proposed a comprehensive process for developing the architectural views in Zachman framework and Ostadzadeh and Nekoui (2009) presented an official language based on Petri Nets, respectively, for the models, which would be investigated for all Zachman framework parts. The suggested model helps the software developers in validation and inspection of all business and unified information technology systems, which would appear in optimizing the affectivity and efficiency of its architecture.

Khayami et al. (2011) developed a method for analysis and evaluation of enterprise architectural plans based on software architecture evaluation knowledge to reach an appropriate and good architecture. Bai (2008) investigated UML diagrams to see they could cover which of C4ISR framework products and prepared the conditions for (Object-oriented Petri Net) OPN simpler production by increasing views to UML diagrams. According to Saldhana and Shatz (2000) state chart diagram, also, one of the UML products was changed to OPN and then a Colored Petri Net of the whole system was presented accordingly. This approach has two phases. In the first phase, some OPN models are presented and in the second phase, these models are connected to each other that end in final model of the system. Mozafari et al. (2011) evaluated the true enterprise architectural behavior by applying official models and making an executive model of enterprise architectural products with Colored Petri Nets.

This paper presents a new method for enterprise architectural products behavioral validation based on C4ISR framework. One of the most important challenges of C4ISR framework is absence of a unified notation for covering all products of different views. Such modeling symbols are necessary because using different modeling languages and symbols for covering the products makes architectures' confusion and non-coordination and makes their works complex and complicated. UML is one of the most current methods of C4ISR products expression. Therefore, an algorithm will be presented to a UML diagram for one of the architectural products mapping. In this paper, among the UML diagrams, the mapping algorithm will be described to sequence diagram. Then, an executive model is presented by Colored Petri Nets for architectural products validation and at the end, the architectural products validation will be described by Colored Petri Nets and by applying CPN TOOLS (Jensen, 1993).

2. Background

2.1 C4ISR Enterprise Architectural Framework

Contrary to Zachman framework, which includes six views, C4ISR framework includes three different views. The three views of this framework show different architecture perspectives. The concept of these three views is different from what is known as view in Zachman framework. The reason for this difference is that, Zachman's framework does not consider the process of work performance while C4ISR framework is mostly based on work performance process. The three standard views of this framework are as follows,

Operational view: This view describes the duties and performances of operational nodes and information circulation among these nodes for operation performance. An operational node is referred to an existence that intervenes in data production, consumption and process related to a mission in some way. Operational elements and nodes, the way of operation performance and supply, the method of information exchange and circulation among nodes will be determined by applying graphic symbols.

System view: This view describes information systems and how they are connected in order to operate performance and supply. In other words, it is a description from the systems and their communication for a work or duty performance or supply. This section describes what shows the role of technology in helping better performance of enterprise missions. By entering from operational view to system view, the operational nodes are replaced with information systems and the intervals are replaced with information transfer lines.

Technical view: This determines the minimum rules complex over the order, performance and communications among the parts or elements of a system codified in order to guarantee the necessities and requirements determined for that system. In fact, the goal of this view is to guarantee the accordance of systems performance with the required expectations.

2.2 Unified Modeling Language (UML)

As the name expresses, UML is a modeling language not a methodology. Usually, each methodology includes a modeling language plus a construction process. Modeling language includes diagrams where each methodology applies for systems design, analysis and demonstration. In summary, UML is a language to describe, to draw, to construct and to document software and non-software systems products of business modeling (Unified Modeling Language Specification Version 1.4.2, OMG, 2004). There are different models for producing executive models from UML diagrams. For example, the executive models can be resulted from behavioral diagrams (such as activity diagram, state chart diagram, sequence diagram, etc.) or from structural diagrams (such as class diagrams, implementation, etc.). In this paper, sequence diagram is applied, which is used to show operation process in a function (Booch, 1999). This behavioral diagram emphasizes on communication model among components and is drawn according to the time of massage sending. The used symbols in this diagram are: life line (massage sending and receiving components), massages, communications, sending and receiving events and also different structures such as order, select, repeat, parallel, etc. shown with composite sections. In this diagram, if the communications are asynchronous, the operation will end by sending the massage and the sender will not wait for operation completion and receiving response. While, in synchronous communications, the sender will wait for the response after sending the massage and the operation will end if the massage sender restarts its performance. The synchronous and asynchronous massages and responses are shown by solid, un-solid and dash arrows, respectively, in sequence diagram.

2.3 Colored Petri Nets

Colored Petri Nets may be used for presenting an executable model. Colored Petri Nets includes nine parts as per the followings (Jensen, 1993):

- $\sum$ is a finite set of non-empty types, also called color sets.
- P is a finite set of places.
- T is a finite set of transitions.
- A is a finite set of arcs such that: $P \cap T = P \cap A = T \cap A = \emptyset$.
- N is a node function. It is defined from A into $P \times T \cup T \times P$.
- C is a color function. It is defined from P into $\sum$.
- G is a guard function. It is defined from T into expressions such that: $\forall t \in T: [Type(G(t)) = B \wedge Type(Var(G(t)) \subseteq \sum]$, where B to denote the Boolean type.
- E is an arc expression function. It is defined from A into expressions such that: $\forall a \in A: [Type(E(a)) = C(p)_{MS} \wedge Type(Var(E(a))) \subseteq \sum]$, where p is the place of N(a).
- I is an initialization function. It is defined from P into closed expressions such that: $\forall p \in P: [Type(I(p)) = C(p)_{MS}]$.

Petri Nets are a graphical tool for officially describing dynamic systems, which have specifications such as concurrency, mutual exclusion and conflict that are of the prominent specifications of distributed environments. CPN Tools are tools for modeling, investigating and analyzing Colored Petri Nets. By applying CPN Tools, we could investigate the behavior of a modeled system by simulation (Jensen, 1993).

3 Proposed Method

3.1 Algorithm of transforming sequence diagram to executive model

The first step for making an executive model is to transform the sequence diagram to Colored Petri Nets. There are different works done in this field (Bernardi, 2002; Ourdani et al., 2006; Emadi & Shams, 2009; Bernardi & Merseguer, 2007). The different structures in sequence diagram including order, selection, parallel and repeating are changed to Colored Petri Nets. In Ourdani et al. (2006) the change is done on the massage sending and receiving object and different massages (synchronous and asynchronous) are changed to Colored Petri Nets in sequence diagram. In this paper, we use the methods used by Emadi and Shams (2009) and Ourdani et al. (2006) that we describe them in the following part.

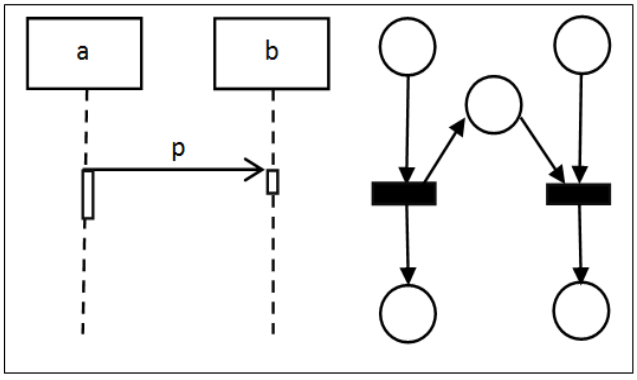

Fig.1. Mapping of an asynchronous message (Ourdani et al., 2006)

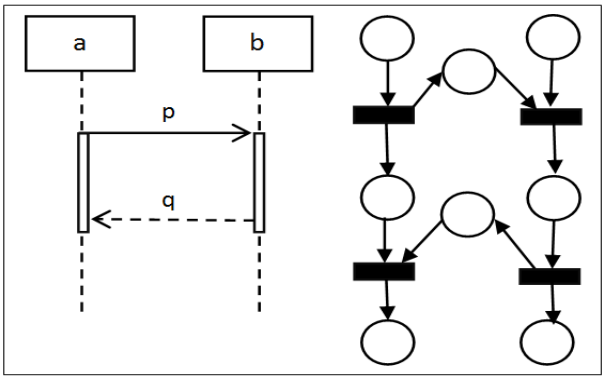

Fig.2. Mapping of a synchronous message (Ourdani et al., 2006)

3.1.1 Translation of Asynchronous message into Colored Petri Nets

Fig. 1 shows translation of an asynchronous message into Colored Petri Net. Such a communication is made up by means of a shared place that is seeing as an outcome place from the sender object and an income place from the receiver object. The sender and the receiver are represented each one as Place-Transition-Place (Ourdani et al., 2006).

3.1.2 Translation of Synchronous Message into Colored Petri Nets

Fig. 2 shows translation of an synchronous message into Colored Petri Net. Such a communication is made up by two shared place that one for call and the second for the return. The sender and receiver are represented each one as P-T-P-T-P (Place-Transition-Place-Transition-Place). The centric P of the P-T-P-T-P sequence plays the part of waiting place for the sender and provided method place for receiver (Bernardi & Merseguer, 2007). The second shared place is equivalent to the acknowledge return or result.

3.1.3 Translation of Synchronous Message into Colored Petri Nets

Fig. 2 shows translation of a synchronous message into Colored Petri Net. Such a communication is made up by two shared place where the first one is for call and the second one is for the return. The sender and receiver are represented each one as P-T-P-T-P (Place-Transition-Place-Transition-Place). The centric P of the P-T-P-T-P sequence plays the part of waiting place for the sender and provided method place for receiver (Bernardi & Merseguer, 2007). The second shared place is equivalent to the acknowledge return or result.

3.1.4 Translation of UML 2.0 Combined Fragments into Colored Petri Nets

A variety of structures like sequence, alternation and option, loop, parallel etc. are presented as combined fragments. In continuation we explain translation of the most popular combined fragments into Colored Petri Nets.

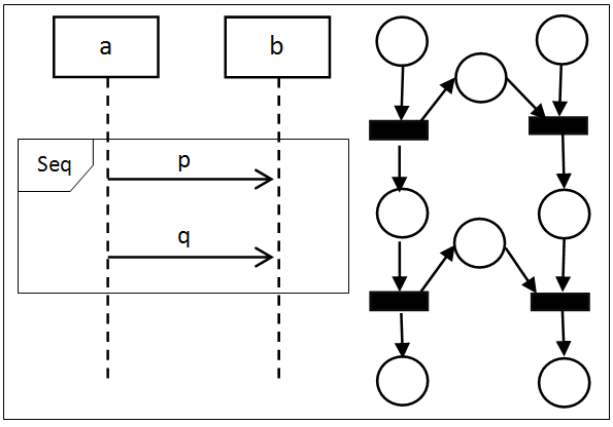

Fig.3. Weak sequencing operator and Colored Petri Net of its equivalent (Emadi & Shams, 2009)

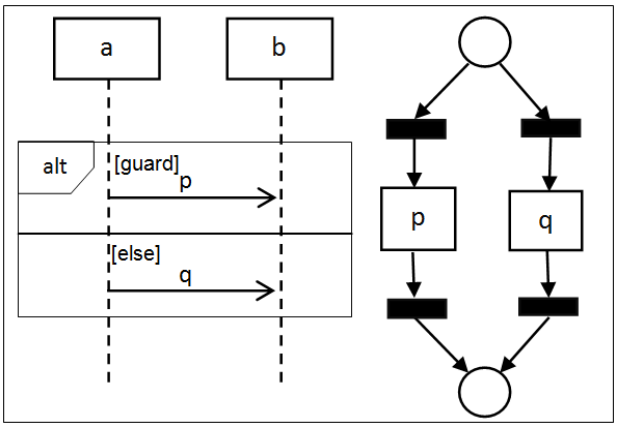

Fig.4. Alt operator and Colored Petri Net of its equivalent (Emadi & Shams, 2009)

- Weak sequence combined fragment: Fig. 3 shows translation of weak sequencing combined fragment into Colored Petri Nets.
- Alternation and option combined fragments: alternation and option combined fragments represent a choice of behavior in sequence diagrams. Alternative and Option operators are denoted **alt** and **opt**, respectively. Fig. 4 shows alt operator and Colored Petri Net of its equivalent.

- Parallel combined fragments: A parallel combined fragment, denoted by **par** operator, represents a parallel merge between the behaviors of the operands. Fig. 5 shows parallel operator and Colored Petri Net of its equivalent.

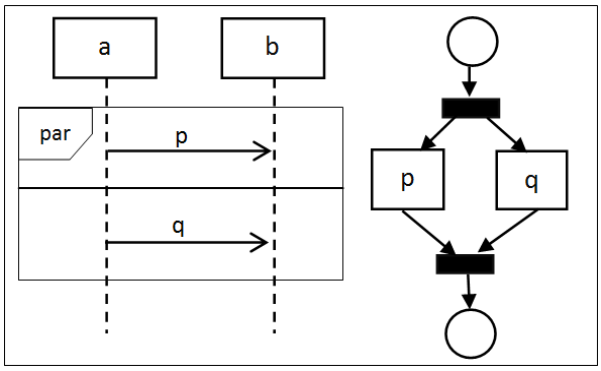

Fig.5. Parallel operator and Colored Petri Net of its equivalent (Emadi & Shams, 2009)

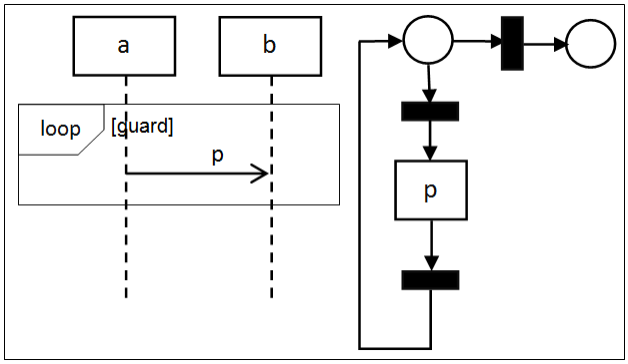

Fig.6. Loop operator and Colored Petri Net of its equivalent (Emadi & Shams, 2009)

- Loop combined fragments: The operator **loop** indicates that the combined fragment represents a repetition structure. The loop operand will be repeated a certain number of times according to the values defined by the designer. Fig. 6 shows loop operator and Colored Petri Net of its equivalent.

3.2 The proposed method for validation of enterprise architecture products behavior

In this part, an executive model is made based on Colored Petri Nets and then a method is presented for validation of the made model. The process of making an executable model for validation of sequence diagram behavior is:

Phase 1: in order to make an executable model by Colored Petri Nets we do chronologically, in the first level, we consider a replaced transition instead of each composed part of the sequence diagram and define a subpage for each replaced transition.

Phase 2: in order to map the sequence diagram to Colored Petri Nets, we use the algorithm presented in the previous part.

Phase 3: we define two event of massage send and receive as C-Send(msg , @r) and C-Receive(msg , @r) for each component (C), that @r shows the performing order of each event. For example, if in Fig. 3, the order of events performance is "at first the massage p was sent by a, then massage p was received by b and then the massage q was sent by a and received by b", then the sending and receiving amount will be quantified as table 1:

Table 1
Send and Receive message in proposed method

a	B
a-Send(p,1)	
	b-Receive(p,2)
a-Send(q,3)	
	b-Receive(q,4)

Phase 4: we express the specifications and requirements of the system defined within the framework in phase 3. We show two examples of systems specifications expressed by presented framework for illustration.

Example. 1: consider, in sequence diagram of Fig. 3, we want to investigate that "component a will not send massage q unless component b receives massage p". The equal phrase is:R(b-Receive(p))

<R(a-Send(q)). In this phase, R(a-Send(q)) is the rank of performance of q massage send by a and R(b-Receive(p)) is the rank of performance of p massage receive by b.

Example. 2: consider, in sequence diagram of Fig. 7, we want to investigate that "at first the component c receives massage p, then the component a receives massage q". The equal phrase is:R(c-Receive(p)) < R(a-Receive(q)).

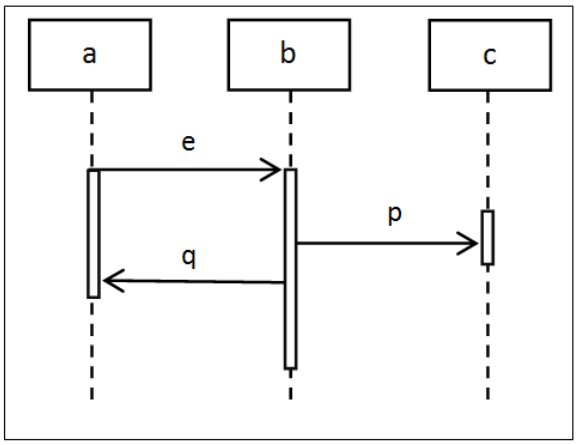

Fig.7.A Sequence Diagram

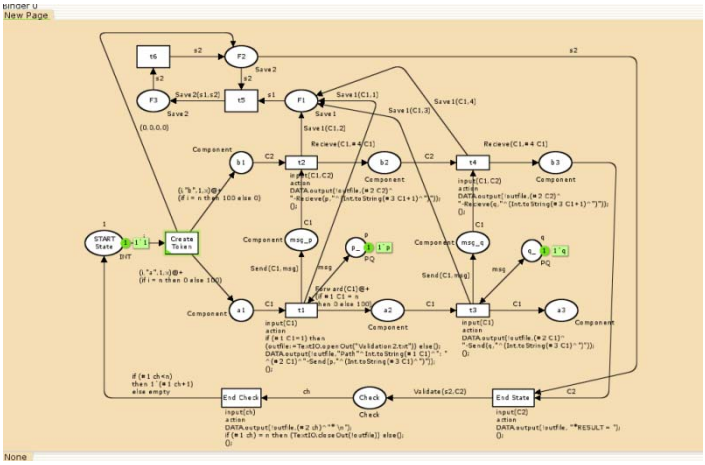

Fig.8. Model made by CPN Tools of Sequence Diagram in Fig. 3

Phase 5: In order to see whether the drawn diagram presents the specifications expressed in phase 4 correctly or not, each specification shall be checked in lieu of all probable executive state in sequence diagram. In other words, in this phase, all the probable executive paths in sequence diagram will be identified and the true behavior of the specifications in each executive path will be investigated and the paths that their true behavior is failed will be identified. In order to apply this parameter to Colored Petri Nets, the initial expression and trusting estimation of input data are necessary affairs. Regarding the fact that CPN Tools uses Standard ML language for data define and modification and this language supports function writing, the operation of checking the true behavior of each path's specification may be performed by writing a function and connecting it to appropriate and show the results in the form of a file and as a report.

3.3 Executable model analysis and simulation results

After making an executable model by Colored Petri Nets, we could perform this model in CPN Tools software and regarding the information in reporting file, see whether the true behavior of architectural products (UML sequence diagram) is confirmed or not.

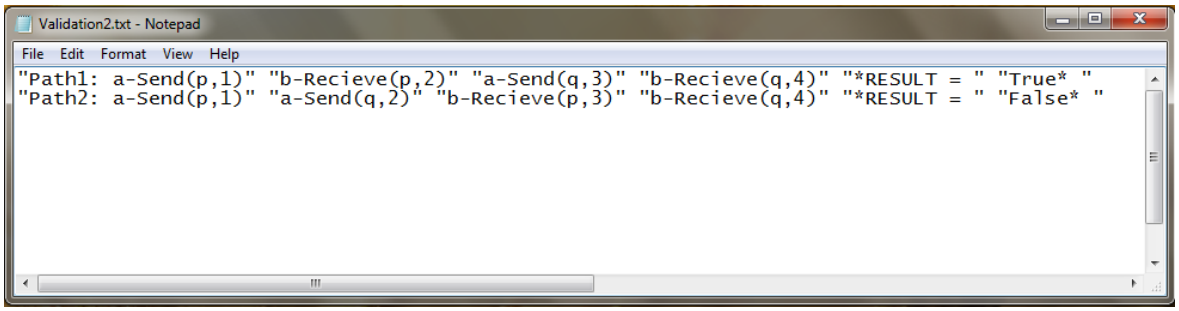

Fig.9. Information resulted from performance of Fig. 8 model

For instance, the made model for Colored Petri Nets of Fig. 3 in CPN Tools is as Fig. 8. In this part, we validate the specification of "component a will not send massage q unless component b receives massage p" for this model, and we could observe the information received upon model performance in the form of a text file in Fig. 9. As it is shown in Fig. 9, all probable executive paths are shown in

sequence diagram of Fig. 3 with the order of events performance. As the true behavior was failed in lieu of one of the paths (RESULT=FALSE), then, this diagram does not present the sequence of the mentioned specification correctly and its true behavior is failed. According to the information, we could also conclude that Path 2 includes a bottleneck.

4 Conclusion

In the present paper, a method has been presented for validation of the products of C4ISR architectural framework with Colored Petri Nets. Therefore, it was attempted to present a model executable by Colored Petri Nets from architectural products. UML diagram was used for expressing architectural products in this paper and sequence diagram was selected among UML diagrams for making a performable model. In order to investigate the true architectural behavior we used sequence diagram symbols i.e. massages, massages send and receive and massages origin and destination and expressed the systems requirements and specifications in a new defined framework on these symbols. In the end, we simulated an executive model made of an example from sequence diagram by applying CPN Tools software and evaluated its true behavior from the results in the form of a file.

References

Bai, X.H.(2008). Study of C4ISR architecture simulation validation with UML and object-based nets.*IEEE 8th International Conference on Computer and Information Technology Workshops, in: Heilongjiang University, Harbin, Heilongjiang, China*.

Bernardi, S., Donatelli, S., & Merseguer, J. (2002). From UML sequence diagrams and state charts to analysable Petri Net Models. *WOSP '02 Proceedings of the 3rd International Workshop on Software and Performance, ACM New York, NY, USA*, 35-45.

Bernardi, S., & Merseguer, J. (2007). Performance Evaluation of UML Design with Stochastic Well-Formed Nets. *The Journal of Systems and Software*, 80, 1843–1865.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison Wesley, Reading MA.

Emadi, S., & Shams, F. (2009). A new executable model for software architecture based on Petri Net. *Indian Journal of Science and Technology*, 2(9), 15-25.

Jensen, K. (1993). An introduction to the theoretical aspects of coloured Petri nets, in: J.W. de Bakker, W.-P. deRoever, G. Rozenberg (Eds.).*A Decade of Concurrency, in: Lecture Notes in Computer Science*, 803, Springer-Verlag, 230-272.

Khayami, R, (2011).Qualitative characteristics of enterprise architecture. *Procedia Computer Science*, 3, 1277–1282.

Locob, M.E., & Jonkers, H. (2005). Quantitative analysis of enterprise architectures. *In the Proceedings of the First International Conference on Interoperability of Enterprise Software and Application (INTEROP-ESA'2005)*, Geneva, Switzerland, 234-248.

Mozafari, M., Harounabadi, A., &Mirabedini, S.J. (2011). A method for validating the behavior of enterprise architecture. *World Applied Science*s, 14(6), 831-841.

Ostadzadeh, S.H., & Nekoui, M.A. (2009). A Petri-Nets based unified modeling approach for Zachman framework cells. *In the Proceedings of ScSS'2009*, 615-618.

Ourdani, A., Esteban, M., Paludetto, & Pascal, J.C. (2006). A Meta Modeling Approach for Sequence Diagram to Petri Nets Transformation within the Requirements Validation Process. *Proceedings of the European Simulation and Modeling Conference,* Toulouse, France, 345-349.

Khayami,R., Tawhidi, A., & Ziarati, K. (2011). Evaluation quality characteristic of enterprise architecture. *In the Proceedings of World Conference on Information Technology*, 3, 1277-1282.

Rezaei, R., & Shams, F. (2008). A methodology to create data architecture in Zachman framework. *World Applied Science Journal*, 3(2), 343-349.

Saldhana, J. A., & Shatz, M. (2000). *UML diagrams to object Petri Net models: An approach for modeling and analysis*, in: Department of Electrical Engineering and Computer Science University of Illinois at Chicago, USA.

Unified Modeling Language Specification Version 1.4.2, OMG.(2004).

Wagenhals, L.W., Haider, S., & Levis, A.H. (2003). Synthesizing executable models of object oriented architectures. *Journal of Systems Engineering*, 6(4), 266-300.