

Enterprise architecture evaluation using architecture framework and UML stereotypes

Narges Shahi^{a*}, Ali Haroun Abadi^b and Hadi aboeei mehrizi^c

^aMA candidate in IT Engineering- Ta'aali Private University, Iran

^bAssistant professor, Department of Computer Science, Tehran Central Branch, Islamic Azad University, Tehran, Iran

^cMA candidate in computer architecture, Department of Computer Science, Arak Branch, Islamic Azad University, Arak, Iran

CHRONICLE

Article history:

Received January 28, 2014

Received in revised format

5 May 2014

Accepted 4 July 2014

Available online

July 6 2014

Keywords:

Performance evaluation

Unified modeling language

Colored Petri networks

ABSTRACT

There is an increasing need for enterprise architecture in numerous organizations with complicated systems with various processes. Support for information technology, organizational units whose elements maintain complex relationships increases. Enterprise architecture is so effective that its non-use in organizations is regarded as their institutional inability in efficient information technology management. The enterprise architecture process generally consists of three phases including strategic programming of information technology, enterprise architecture programming and enterprise architecture implementation. Each phase must be implemented sequentially and one single flaw in each phase may result in a flaw in the whole architecture and, consequently, in extra costs and time. If a model is mapped for the issue and then it is evaluated before enterprise architecture implementation in the second phase, the possible flaws in implementation process are prevented. In this study, the processes of enterprise architecture are illustrated through UML diagrams, and the architecture is evaluated in programming phase through transforming the UML diagrams to Petri nets. The results indicate that the high costs of the implementation phase will be reduced.

© 2014 Growing Science Ltd. All rights reserved.

1. Introduction

Enterprise architecture (EA) is a comprehensive technique for explaining the current or future structure and behavior of an organization processes, information systems as well as organizational sub-units. Enterprise architecture is so important that the lack of it in organizations may result institutional inability in efficient information technology management. The enterprise architecture process generally includes three phases including strategic information technology programming, enterprise architecture programming and enterprise architecture implementation. If an architect can evaluate the architecture in the second phase, the designed architecture is assessable from various perspectives and it can be reformed by the architect. Creating an implementable model of architecture is a method for architecture evaluation, regarding a formal description of the given architecture, and through which, before implementation, the final behavior of the system is observable and assessable.

*Corresponding author.

E-mail addresses: shahinarges@yahoo.com (N. Shahi)

Therefore, the probable flaws in the architecture can be traced, properly. This article aims at using a real-world model to explain the enterprise architecture. The features of enterprise architecture are demonstrated through UML diagrams, and the necessary information relevant to non-functional requirements are annotated in operational profiles format in terms of stereotypes and labels to the diagrams, which could be transformed and evaluated into an implementable model based on colored Petri net. Transforming the real-world model to implementation model makes it possible to evaluate the enterprise architecture efficiency in an applicable model. In the second part of the paper, the field of the work is discussed; the third part describes the activities relevant to the topic of this paper. Part four indicates use of Colored Petri nets in Creating Performance Model and in The fifth part, we describe a suggested solution for presenting a formal model. The sixth part presents a case study and the part seven presents compares the proposed method with the former ones. Finally, the last part is assigned to the conclusions.

2. Operation field

The operation field includes enterprise architecture, Unified modeling language, Petri nets, and colored Petri nets.

2.1. Enterprise Architecture

Enterprise architecture (EA) is the definition and presentation of a high-level perspective toward organizational business processes and IT systems, their mutual relationships, and how much such processes and systems are shared through various parts of an organization. The primary purpose of EA is to define the desirable future status of organizational business processes and IT systems, which mostly creates a path map of the current status to achieve such purpose (Tamm et al., 2011).

2.2. Enterprise Architecture Framework

The architecture includes a large number of documents describing all parts of the organization. The problem facing such documents is how to include and hoe to use them, properly. Consequently, to arrange and to describe enterprise architecture, a framework is needed (Afshani et al., 2012). Enterprise architecture framework (EAF) created a direction for developing various architectures, organizing architecture models and an architecture that manages the tasks within the organization, and making a connection to develop complicated organizational structure (Kim et al., 2005). In enterprise architecture process, the architect can make use of the framework as a regulator for structure. Some people developed diverse frameworks such as Zachman, C4ISR, FEAF, TEAF, etc. among which the C4ISR framework is utilized due to its possibility of expressing by graphical models (Afshani et al., 2012).

2.3. Unified Modeling Language

UML as a Unified modeling language supporting object-oriented concepts, was proposed in 1990s. It has numerous functions and, after only one year, was accepted by object management group (OMG), as the standard modeling language. Since UML is not a graphical model, assessment of software systems is impossible. However, OMG, responsible for developing UML, introduced a profile to support performance concepts (Haroonabadi et al., 2013). The object managing group proposed UML as an architecture describing standard language in which the system components description, their visible features and the connections among components are illustrated. However, none of these descriptions is able to assess non-functional attributes, and they must be transformed into an implementable model through to assess non-functional attributes (Emadi & Shams, 2009).

2.4. Petri Nets

Petri Nets was originally proposed by Carl Adam Petri in the beginning of 1962. Petri nets, as a mathematical and graphical tool for modeling computer systems, were introduced. Due to their

simplicity and high capability, Petri nets are utilized largely in enterprise architecture assessment and creating an applicable model (Emadi & Shams, 2008). Such nets express system data currents through circular places, transfer rectangle, directional arcs and spot tokens (Fang et al., 2013).

2.4.1. Colored Petri Nets

Petri nets create a graphical symbol for modeling the systems and the analysis. Colored petri nets (CPN) integrate the strengths of common petri nets with a high-level programming language, and makes them more proper for great systems. CPN model is a show executable of a system that can be analyzed through simulation (Gehlot & Nigro, 2010).

3. Related Works

Saldhana and Shatz (2000) developed a method for creating a formal model of UML diagrams by presenting a methodology for creating a formal model of the system for analyzing and behavioral modeling. In fact, the state diagram of UML and collaboration diagrams of UML were transformed into Object Petri net. Then, through the presented formal model, correct-finding of behavioral features of UML was investigated to discover the synchronization-based behaviors such as deadlock.

Shin et al. (2003) proposed a method for creating a formal model of UML diagrams and suggested a process for transforming UML-based system model into CPN model. The UML-based system model, according to the connection among use case, class model and collaboration models was written in CPN model having hierarchical structure.

Afshani et al. (2012) proposed a new format based on fuzzy UML concept based on fuzzy theory for some of C4ISR outcomes such as Logical Data Model (OV-7) and Operational Event/Trace Description (OV-6c), Systems Event/Trace Description (SV-10c). Their recommended model led to the creation of a number of outcomes of C4ISR framework. They also evaluated a couple of metrics including architecture performance and the security of the enterprise architecture net. Other types of qualitative metrics such as dependability and validity were not evaluated. Haroonabadi et al. (2008) discussed the behavioral description of uncertain systems from fuzzy status diagram. They also studied the role of behavioral diagram in F-UML in the process of software function modeling. Lima et al. (2009) proposed a method for evaluating the correction of the behavior and the validity of the sequence diagram of UML. This method uses the source and the destination of messages in sequence diagram and such diagrams are written in Promella language. Then the SPIN tool is used for simulation. In neither cases of the above, any study has accomplished on the use case diagram, activity diagram, component diagram with performance stereotype.

4. The Use of Colored Petri nets in creating performance model

To create a CPN model, the UML diagrams including performance annotation are supposed as the input. The use case diagram illustrates the working load of the system. The component diagram presents a design of the software resources of the system, and it is utilized to illustrate the static system structure. The action diagram indicates the service request of the resources in terms of different operations. The annotation on the above mentioned diagrams include stereotypes and labeled amounts. The outputs of the architecture are collections of graphical, textual and table models describing the architecture (Afshani et al., 2012). This paper focuses upon performance assessment of enterprise architecture through UML stereotypes, and describes some of C4ISR outputs such as logical data model (OV-1), (OV-2) and (OV-5).

4.1. The Stereotypes in Use Case Diagram

In general, each user in use case indicates a sequence of requests in system; this diagram includes following stereotypes:

<<PAopenload>> is applied when the sequence of the requests is unlimited; its tags are: PAoccurrence that indicates the interval between two sequential requests.

<<PAclosedload>> is used when the sequence of the request is limited, and it includes the following tags:

PApopulation indicates all the requests within the system. PAextDelay indicates the interval between a request and the following interaction to the system (Khamseha et al., 2013).

The high level operational concepts diagram (Ov-1) is the most general output of the architecture description and possesses the most flexible format (Haroon Abadi et al., 2013). The use case provides a way for describing the operational concept (Bai, 2008)

4.2. The Stereotypes in Activity Diagram

In activity diagram, the activities announce their service request from a system resource through <<PAstep>>, and include the following labels:

PAhost is used for determining active resources. PADemand determines the service request.

The activity diagram can describe the operational activities relevant to the operational activities model of the architecture C4ISR, (Ov-5) and the data or the information transferred among operational nodes (Bia, 2008).

4.3. The Stereotypes in the component diagram

<<PAhost>> is used for determining the components and it can model an active resource; it is also a design of the resources available in the system. PARate shows the process speed. PASchdpolicy illustrates the timing policy (Emadi & Shams, 2008). Component diagram is a method to describe (Ov-2).

Table 1

The stereotypes and the labels of the UML diagrams

UML Diagram	Element in UML	Stereotype	Tag
Use Case Diagram	Actor	<<PAopenLoad>>	PAoccurrence
		<<PAclosedLoad>>	PApopulation PAextDelay
Activity Diagram	Action State	<<PAstep>>	PAhost PADemand
Component Diagram	Component	<<PAhost>>	PASchdpolicy PARate

5. The Recommended Algorithm

CPN model consists of T transfers and C colors representing the number of client classes. The transfers illustrate UML model resources in CPN. An extra transfer in CPN is defined through the <<PAclosedLoad>> for each agent, indicating the time consumed for the finishing of an interaction to the system and the beginning of its next repetition (such as the user thinking time). Numerous factors have various working loads in system. In this case, CPN model can be divided into several CPN sub-models separated from each other. Each sub-model will possess its own working load. The requests of a sub-model can have several classes. The classes available in each sub-model are illustrated by different colors in the CPN model.

In open Petri net, the input enters into transfer, where the initial action of the activity diagram use it. The rate of entering to the Petri net equals the PAoccurrence in an agent that uses activity diagram for implementing its use case.

Several actions can use one resource. To support this issue in CPN model the following attribute is defined. assuming that the system resources are shown as Eq. (1)

$$RES = \{res_1, res_2, \dots, res_T\} \quad (1)$$

Counter [res] shows the sum of the actions requested service from res. we define a unique index for any resource by identity

$$identity [res_i] = i \quad \text{for each } res_i \in RES \quad (2)$$

The set of the actions using res resources is as follows Eq. (3):

$$ACT = \{act_1, act_2, \dots, act_m\} \quad (3)$$

For each resource $res \in RES$

We label all the action in the set $\{act \in ACT \mid resource(act) = res\}$

Through one unique number in the $[1, 2, \dots, counter[res]]$ interval. The unique number is specified by indicator[act].

5.1. The Algorithm of Transforming the UML Model into the CPN Model is as follows:

It is assumed that the X agent is annotated by <<PAopenload>>.

1) the amount of attributes in Eq. (4) are calculated:

$$counter [res] \quad \forall res \in RES, \quad (4)$$

$$indicator [act] \quad \forall act \in ACT,$$

$$C = \max_{res \in RES} \{counter [res]\}$$

$$T = |R|$$

2)The amount of process rate and arrival rate are calculated in Eq. (5) as follows,

$$SR = rate [r] / demand [act] \quad (5)$$

The $\lambda[r]$ indicating the rate of a client with r class is illustrated in Eq. (6) as follows,

$$\lambda[r] = arrivalrate[X]. \quad (6)$$

If the X agent is annotated by <<PAopenload>>, the following changes are applied to the algorithm:

The number of the transfers is one more than available resources in component diagram.

$$T = |RES| + 1 \quad (7)$$

Extra transfer is shown by '0' for indicating the delay. The fire rate of this transfer for all classes is stated in Eq. (8) as follows,

$$SR [0 , s] = 1 / \text{extdelay} [X] \tag{8}$$

According to the PApopulation label, N is the number of requests available in system and it is determined as Eq. (9) as follows,

$$N = \text{population}[X]. \tag{9}$$

Integrating the mentioned algorithms, a new algorithm is made in which the number of the CPN sub-models equals $A+B$. A is the number of agent with the $\ll\text{PAopenLoad}\gg$ stereotype and B indicates the number of agent with the $\ll\text{PAClosedLoad}\gg$ stereotype.

6. The Case Study

The case study includes the process of getting cash from ATM. Fig. 1 illustrates the use case diagram only for one use case. Fig. 2 indicates component diagram and Fig. 3 shows activity diagram. This activity diagram has 8 acts. In addition, Fig. 3 shows details of activity diagram.

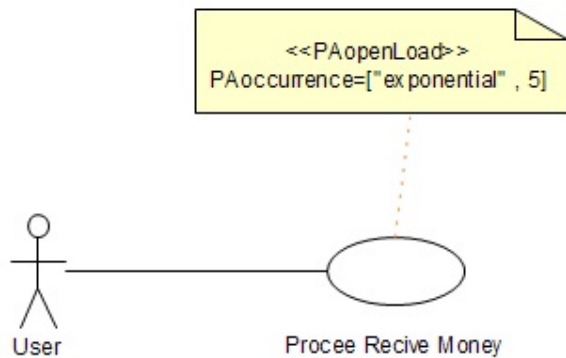


Fig. 1. use case relevant to action diagram

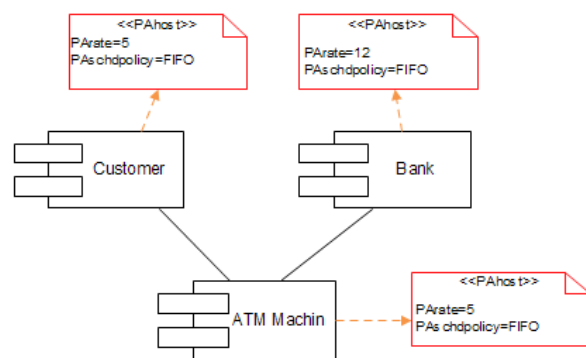


Fig. 2. The component diagram with annotation

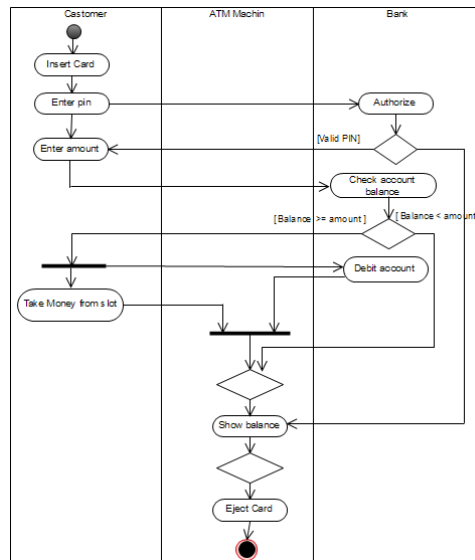


Fig. 3. Activity diagram

Fig. 4 demonstrates the middle graph which determines the resources and the number of their used times. One node is regarded for each act; the resources are numbered. Each repetition in the use of a resource adds one unit to indicator. Fig. 5 illustrates the colored Petri net relevant to the middle graph. If the use case relevant to the activity diagram has the following label:

PAoccurrence=[“exponential”,5]

Then: $\lambda=1/5$

“PAdemand”, for the actions using the resource 1, has the following amounts (the Petri net has 4 classes of colors):

{0.2 ,0.1 ,0.5,0.3}

The fire rate in the Petri net transfer for the resource 1 is as follows,

$$SR=rate[res]/demand[act] =(25,50,10,16/6)$$

Though determining arrival rate and fire rate, Petri nets are analyzed. CPN Tools is simulation tool of Petri nets and Rational Rose tool is the implementing tool of the UML diagrams.

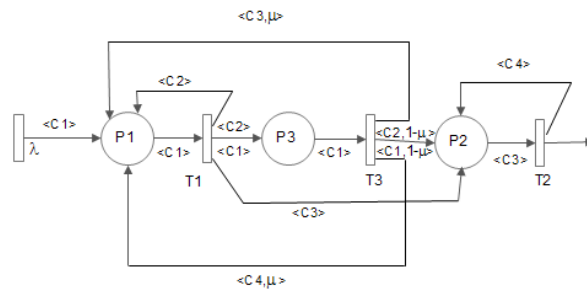
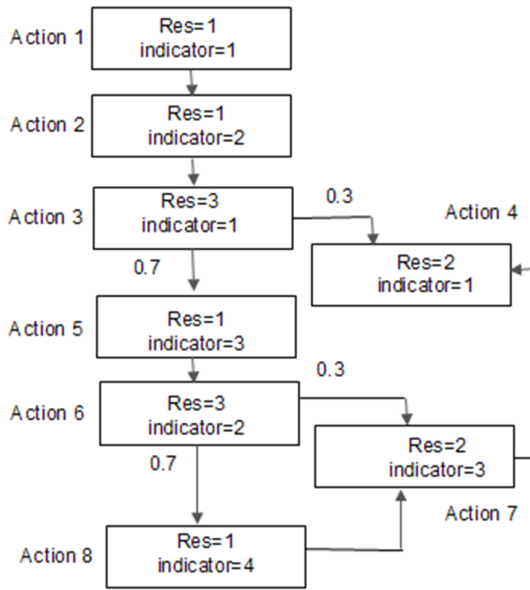


Fig. 4. The middle graph between real model and formal model

Fig. 5. The Petri net relevant to the action Diagram

7. Comparing the proposed method with the former ones

Table 2 demonstrates the features of the proposed method compared with former methods. The recommended method, in addition to the abilities of those methods, is able of evaluating the enterprise architecture through software resources; in fact, it is the unique ability of this recommended model.

Table 2

Comparing the recommended method and the former methods

Assessment criterion	Models			
	Archimate	Levis	OSAN	Suggestive
implementation	no	Yes	Yes	Yes
Value-oriented support	no	Yes	Yes	Yes
Efficiency assessment	Yes	Yes	Yes	Yes
Assessment of correction in architecture behavior	no	Yes	no	no
Utilizing the software resources	no	no	no	Yes

8. Conclusions

As mentioned earlier, Enterprise architecture is so effective that its non-use in organizations was regarded as their institutional inability in efficient information technology management. This paper

proposed new method for evaluating enterprise architecture based on a formal model (Colored Petri Nets) to evaluate the non-functional requirements. As a result, before implementing the enterprise architecture, a model of the issue was designed and evaluated to prevent the Probable flaws in the implementation phase.

References

- Afshani, J., Haroon Abadi, A., & Abbasi, M. (2012). Enterprise Assessment using Fuzzy Colored Petri Nets. MA in Computer engineering, software; Islamic Azad University. *Khoozestan Science and Technology Branch*.
- Afshani, J., Harounabadi, A., & Dezfouli, M. (2012). A new model for designing uncertain enterprise architecture. *Management Science Letters*, 2(2), 689-696.
- Bai, X. H. (2008, July). An application with UML object-based PETRI NETS for C4ISR architecture simulation validation. In *Machine Learning and Cybernetics, 2008 International Conference on* (Vol. 4, pp. 2257-2263). IEEE.
- Emadi, S., & Shams, F. (2008). From UML component diagram to an executable model based on petri nets. In *Information Technology, 2008. ITSIM 2008. International Symposium on* (Vol. 4, pp. 1-8). IEEE.
- Fang, Z., DeLaurentis, D., & Davendralingam, N. (2013). An Approach to Facilitate Decision Making on Architecture Evolution Strategies. *Procedia Computer Science*, 16, 275-282.
- Gehlot, V., & Nigro, C. (2010, December). An introduction to systems modeling and simulation with colored Petri nets. In *Simulation Conference (WSC), Proceedings of the 2010 Winter* (pp. 104-118). IEEE.
- Haroonabadi, A. (2013). Assessment of data systems through Fuzzy Colored Petri Nets. 16th National Annual Conference of Iran's Computer Group. Sharif University of Technology, Tehran.
- Haroonabadi, A., Teshnehlab, M., & Movaghar, A. (2008). A novel method for behavior modeling in uncertain information systems. *World Academy of Science, Engineering and Technology*, 41, 959-966.
- Emadi, S., & Shams, F. (2009). Presenting an implementable model of the software architecture for assessing the efficiency. *Iranian Computer Scientific Journal*, 7(1).
- Kim, J. W., Kim, Y. G., Kwon, J. H., Hong, S. H., Song, C. Y., & Baik, D. K. (2005, August). An enterprise architecture framework based on a common information technology domain (EAFIT) for improving interoperability among heterogeneous information systems. In *Software Engineering Research, Management and Applications, 2005. Third ACIS International Conference on* (pp. 198-205). IEEE.
- Khamseha, S., Sayyadi, F., & Yektaie, M.H. (2013). A new method for performance evaluation of enterprise architecture using stereotypes. *Management Science Letters*, 3, 2681-2690.
- Lima, V., Talhi, C., Mouheb, D., Debbabi, M., Wang, L., & Pourzandi, M. (2009). Formal verification and validation of UML 2.0 sequence diagrams using source and destination of messages. *Electronic Notes in Theoretical Computer Science*, 254, 143-160.
- Saldhana, J., & Shatz, S. M. (2000, July). Uml diagrams to object petri net models: An approach for modeling and analysis. In *International Conference on Software Engineering and Knowledge Engineering* (pp. 103-110).
- Shin, M. E., Levis, A. H., & Wagenhals, L. W. (2003, October). Transformation of UML-based system model to design/CPN model for validating system behavior. In *Proc. of the 6th Int. Conf. on the UML/Workshop on Compositional Verification of the UML Models*.
- Tamm, T., Seddon, P. B., Shanks, G., & Reynolds, P. (2011). How does enterprise architecture add value to organisations. *Communications of the Association for Information Systems*, 28(1), 141-168.