

## Evaluation of software architecture using fuzzy color Petri net

Zohreh Shiriyani Dehkordi<sup>a\*</sup>, Ali Harounabadi<sup>b</sup> and Saeed Parsa<sup>c</sup>

<sup>a</sup>Department of Computer, Shabestar Branch, Islamic Azad University, Shabestar, Iran

<sup>b</sup>Department of Computer, Central Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>c</sup>Department of Computer, Tehran Branch, Science and Industry University, Tehran, Iran

### CHRONICLE

#### Article history:

Received October 25, 2012

Received in revised format

12 December 2012

Accepted 16 December 2012

Available online

December 17 2012

#### Keywords:

Unified Modeling Language

Color Petri Net

Fuzzy-UML

Fuzzy- Color Petri Net

### ABSTRACT

Unified modeling language is a semi-formal and standard language for describing software architecture easily used to meet requirement functionality and to describe behavioral and structural specifications in software engineering. However, since UML is not a formal model, direct evaluation of software architecture is not possible, directly. On the other hand, evaluating software architecture in initial phases helps us consider more requirements and it reduces designing expenditures. Therefore, converting pragmatic model to formal model is necessary for software architecture evaluation. For evaluating software architecture and making an executive model, using color Petri net are useful because of simplicity, high capability and enhancing theoretical mathematics. Using language variables and fuzzy logic causes a higher accuracy in computations associated with efficiency evaluation. In this paper, we present fuzzy logic and reaction analysis before implementation, which can improve efficiency of design problems.

## 1. Introduction

Unified modelling language (UML), as a modelling language, which supports objective concepts was propounded in 1990s. Nowadays, UML has spread abundantly and there are literally tremendous applications of UML in various industries but the primary concern is that most UML applications deal with some sort of uncertainties (Ma, 2005). Therefore, a new version under the name of Fuzzy-UML has been provided by inserting uncertainty in UML, which allows better exploiting this language and considers both aspects of behaviour and structure. Propounded idea in this article is that we can explain software architecture by a pragmatic model. Then some features from software architecture are presented by UML diagrams and necessary information associated with non-functionality needs are written as annotation in terms of operational profiles, stereotypes and tags. By this way, it is possible to convert an executive model based on fuzzy color Petri net and perform an evaluation.

\*Corresponding author.

E-mail addresses: z.shiriyani@yahoo.com (Z. Shiriyani)

Both structural and dynamic must be considered in evaluation of software architecture. Application of use case diagram and deployment diagram are implemented to show structural aspects and diagrams of activity, sequence and state are used for behavioural aspects. In this paper, we study Fuzzy-UML to show uncertainty in system. We implement fuzzy sequence diagram to demonstrate behavioural specifications. In the second section of article the condition of doing work is considered. In third section, related works are studied based on topic of the article. Proposed method for color Petri net, fuzzy data diagram, fuzzy sequence diagram and analysis are considered in the fourth section. Fifth section is associated with evaluation and practical results and sixth section summarizes conclusion.

## **2. Background**

In this section, we state some necessary definitions on UML software architecture, fuzzy logic and color Petri net are each item is considered, briefly.

### *2.1. Fuzzy Logic*

Fuzzy logic was defined in 1974 and the theory has been expanded including fuzzy sets theory, fuzzy logic and fuzzy measurement. Theory of fuzzy set is an extension of traditional fuzzy logic and fuzzy measurement in an extension of probability measurement.

### *2.2. Petri Net*

Karel Petri in 1962 is believed to be the first who introduced the idea of Petri nets and it has been considered as powerful tool for synchronic modeling. Using these nets for evaluating software architecture and creating executive model are considered for simplicity and high efficiency. Petri nets diagrams are demonstrated and the concepts and structure have been defined.

#### *2.2.1. Color Petri Nets*

Color Petri nets have been introduced as a developed model from Petri nets. Some concepts like “expression”, “gard” and “color” are introduced in addition to places, transition and token. Color Petri nets use capabilities of simple Petri nets and programming languages. Data quantities are carried by token in these nets. Color Petri nets provide more accurate models from complicated non-cophase processed systems. In these networks, token are distinguishable and this is on the contrary to Petri nets since each of these token has some features like “color”. Color Petri nets with color token have a high capability for providing concept of convergent process.

#### *2.2.2. Fuzzy Petri Net*

Petri nets provide a powerful and formal modeling method based on a mathematical structure. These nets have diagramic representation for system models as net diagrams but they cannot solve uncertain problems. Therefore, a new tool named Fuzzy Petri net can be useful. Fuzzy Petri net is a compound fuzzy set theory with Petri net. Place, token and transition are elements from Petri net that fuzzy making is possible on them. Fuzzy token explains the quantity of belonging token to the place. This quantity is  $\{0,1\}$  in usual Petri net, but in Fuzzy Petri net every number can be between  $[0,1]$ .

## **3. Related Works**

Nowadays, there have been tremendous works on converting a pragmatic model to a formal model. Most of the activities are associated with Petri net and queue net (Balsamo et al., 2004; Pettit & Gomaa, 2004). Some of them also use formal models and state diagram based on UML diagram (Saldhana & Shatz, 2000). Among other activities, Merseguer and Campos (2004) introduced the idea

of stochastic Petri net and the other work was dedicated to object Petri net. Merseguer and Campos (2004) reviewed sub-profile of efficiency standard by object management group and considered the role of UML diagrams in software efficiency. Saldhana and Shatz (2000) converted state diagram to Petri net of object and provided a model of colored Petri net for the entire system. Emadi (2008) suggested that we can describe information about efficiency and reliability of software systems by attaching stereotypes and tags to UML diagrams and then an executive model from software architecture are provided to convert them to stochastic color Petri net. This net is described by a common grammar and CPN tools analyze software architecture.

According to Harounabadi (2010), it is possible to describe architecture by Fuzzy UML (F-UML) diagrams and adding efficiency information by stereotypes and tags. We can make an executive model from software architecture by converting these charts to fuzzy color Petri net. Finally, software efficiency is evaluated based on CPN tools and the results are compared in terms of some diagrams as fuzzy and non-fuzzy with emphasis on meters of sources operation and queue length.

Motmeni (2009) stated that we could convert diagram of fuzzy state, which is one of the available diagrams in modeling F-UML to fuzzy color Petri net by an algorithm. A diagram of fuzzy state describes a static aspect of software architecture in uncertain systems and its converting to fuzzy Petri net makes an executive model from software architecture, which helps structural analysis of software architecture.

Motmeni (2004) stated that we could convert diagram of fuzzy activity to fuzzy Petri net by an algorithm. This diagram is implemented to demonstrate available actions in fuzzy uncertain systems, which indicate dynamic aspect of software architecture and Petri net resulting from and it is used for analyzing and evaluating method of execution time in software architecture. In different research groups, evaluation of efficiency requirements in early level of software development and in the level of software architecture has been considered and many methods have been suggested too. However, none of the above articles has considered using efficiency sub-profiles in analysis of fuzzy Petri net for efficiency evaluation.

#### **4. Proposed Method**

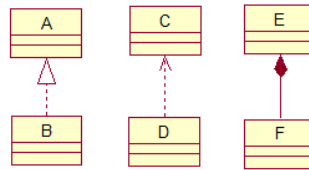
The primary objective of this paper is to provide a fuzzy formal model for F-UML diagrams and to analyze the system before implementation stage. First, we extract some features from software architecture and their stating will be considered based on pragmatic model using UML. Then, we convert UML to fuzzy-UML based on uncertainty, which is considered in software systems. Finally, we convert F-UML to fuzzy Petri net and evaluate, too.

##### *4.1. Explaining Software Architecture Based on UML Diagrams*

According to Pour Kamali (2001), the main structure of software architecture includes module structure, component-connector structure and allocation structure. UML is used for explaining main structure of architecture.

##### *4.1.1. Modeling Module Structure by UML*

Module structure includes a set of modules beside each other with intermediaries and their connections. These connections can be considered as aggregation, dependency and generalization demonstrated in Fig. 1 where module visions are described by UML, very easily. First, systems' module will be identified by use case diagram and then module structure will be implemented by activity diagrams.



**Fig.1.** Symbol Relationship Aggregation, Dependency and Generalization

#### 4.1.2. Modeling Component and Connector by UML

To illustrate component and connector structure including synchronization visions, process and common data, representing running time of system, the way of processes cooperation or different kinds of processes connection is necessary and collaboration or sequence diagrams are useful to show this structure. Sequence diagram is incorporated for demonstrating trivial acts and synchronization operations must be presented to indicate synchronization or decision making about source. In addition, we can use synchronization levels in activity diagram to show these cases.

#### 4.1.3. Modeling allocation structure by UML

Allocation structure normally incorporates three sub structure including deployment, implementation and allocation of tasks and UML deployment diagram generally presents deployment structured, completely. In UML, deployment diagram is a diagram with some nodes connected to each other by link lines. Nodes illustrate components, which show running or operation of that component on the node. Some components with dotted line arrows are jointed, which shows using a component from services of the other component. Components diagram is implemented to demonstrate implementation visions and use case diagram is applied to demonstrate visions of tasks allocation.

#### 4.2. Converting UML to fuzzy-UML

UML as a standard language of objectivism can consider structural and behavioral aspects in software engineering. A developed version named Fuzzy-UML has been presented by considering uncertainty to UML, which includes both sides of structure and behavior in the system.

##### 4.2.1. Fuzzy data model

According to Harounabadi (2007), to determine system behavior, its functionally requirement must be specified and various services are important for us based on objectivism. Here, each service is called user case. To identify a system, we must specify which services are provided and received by system and this helps identify needs, more precisely. If a service is presented uncertainly, then fuzzy use case will be propounded. A symbol used in Fuzzy-UML is shown in Fig. 2 as follows,

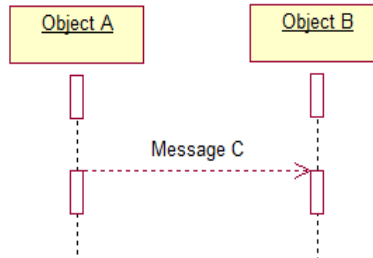


**Fig.2.** Symbol Fuzzy use case in Fuzzy-UML

Use case is a sequence of operations performed by a system to incorporate an observable and valuable result for a user. In other words, for each use case we can set some scenarios to show process of providing service for use case.

### 4.2.2. Fuzzy sequence diagram

According to Harounabadi at (2007) sequence diagram is incorporated for implementing use case in UML. If there is uncertainty with use case then sequence diagram will be uncertain, too. Dotted lines are implemented to show uncertain message in the mentioned diagram message in sequence diagram demonstrate methods of the object under study and uncertainty in the method contains two levels of fuzzy making which are as follows: Method belongs to the object with degree of possession and nature of method is as fuzzy.



**Fig. 3.** Symbol Fuzzy Message in Fuzzy Sequence Diagram

### 4.3. Making a Fuzzy Color Petri Net

According to Kumar and Jaspersnet at (1992) we consider UML diagram including annotation of efficiency as input for making model F-CPN. Diagram of use case demonstrates work load of system. Deployment diagram presents a plan of physical researches and sequence diagram states sequence demand from resources for various activities. Annotation writing on the above diagrams incorporates stereotypes and values with tags. Each actor in use case diagram incorporates a sequence of demands. If the sequence is open, then the actor will be specified with this stereotype: <<PAopenLoad>> and if the sequence is closed then the actor will be specified with this stereotype: <<PAClosedLoad>>. In an actor with stereotype: <<PAClosedLoad>> we have a tags named PApopulation, which demonstrates total number of available demands in the system.

PAextDelay presents time span between a completed demand and next cooperation with system. We have PAoccurrence in actor with stereotype <<PAopenLoad>>, which shows time between two successive demands. This actor maintains, at least, one use case and details of this service (use case) are demonstrated by sequence diagram and activity diagram. Note that there are two important aspects in modeling efficiency of computer systems: Synchronization among different synchronic activities contention for resources. These above feature are reasons for using Petri net and its preference to net of support queue (if queue net supports second case).

**Table 1**  
Stereotypes and Tags Coordinate on UML (Kumar & Jaspersnet, 2012)

UML Diagram	Element in UML	Stereotype	Tag
Use Case Diagram	Actor	<<PAopenLoad>> <<PAClosedLoad>>	PAoccurrence PApopulation PAextDelay
Sequence Diagram	Message	<<PAstep>>	PAhost PAdemand PAprob
Deployment Diagram	Node	<<PAhost>>	PAshdpolicy PARate

A source with some acts can be used in the system; therefore, rate of service (fire) is different for acts. In color Petri net different colors, which are related to tokens show this topic.

#### 4.4. Proposed Algorithm

F-CPN model includes T transition and C (different color), which demonstrates the number of clients classes. Transitions in F-CPN present resource of model UML. For each actor with stereotype <<PAclosedLoad>> an additional transmission is described in F-CPN, which shows time spent between end of a cooperation with system and start of the next repetition (for example user thought). Demands belonging to a sub-model can have some classes. Available classes in each sub-model are demonstrated with various colors in model F-CPN. Available color in each sub-model is considered to be exclusive for that sub model. An open Petri net is a new one, which has input/output to external environment. Closed Petri net does not contain any relationship with external environmental, too.

Open and closed Petri nets support stereotype <<PAopenLoad>> and <<PAclosedLoad>> in the diagram of use case, respectively. Note that a source with acts can be implemented in the system. Next, we describe the following features for supporting the proposed F-CPN.

Since a system includes resources we show them with  $RES = \{res_1, res_2, \dots, res_m\}$  where a feature named counter [res] is defined for each resource of  $res \in RES$ . Counter [res] shows a set of demands where the service requests from res. We define a special indicator with the help of identity for identify resources.

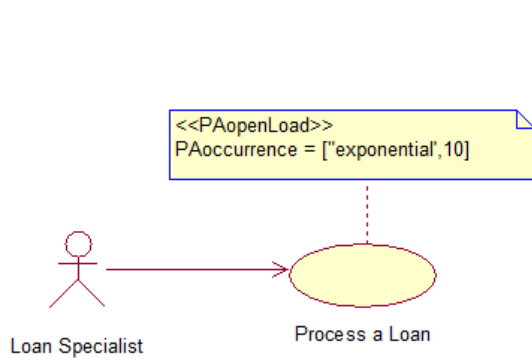
We demonstrate a set of acts which use resource *res* with  $ACT = \{act_1, act_2, \dots, act_m\}$ . Therefore, we have: Counter [res]=m. For each resource  $res \in RES$ , We tags all demands in set  $\{act \in ACT \mid resource(act)=res\}$  as a special number in span  $[1, 2, \dots, counter[res]]$ . This number is specified with feature indicator [act]. Input in open Petri net enters transition part, which uses that resource for initial demand of fuzzy sequence diagram. Arrival rate to fuzzy Petri nets is equal to PAoccurrence in an actor used fuzzy sequence for implementing use case. Transition probability in fuzzy Petri net is resulted from PAprob in transition of fuzzy sequence diagram. Algorithm for converting F-UML to F-CPN is as follows for evaluation of software efficiency (on the assumption that actor X has been written as a annotation with <<PAopenLoad>>).

1. *Values of the following features are specified:*  
*counter [res]  $\forall res \in RES$ ,*  
*Indicator [act]  $\forall act \in ACT$ ,*  
*C= maximum  $res \in RES \{counter [res]\}$  and T*
2. *Probability matrix will be computed.*
3. *SR[i, r](Service rate for clients with class r in transition i) is computed. (m is an act in fuzzy sequence diagram):*  
*SR[i,r] = rate [r] / demand [act]*  
*Where i = identity [ resource [m] ] , r = indicator [act]*
4. *Vector  $\lambda [r]$  which shows arrival rate for clients with class r, is specified like this:*  
 *$\lambda[r]= arrival rate [x]$*

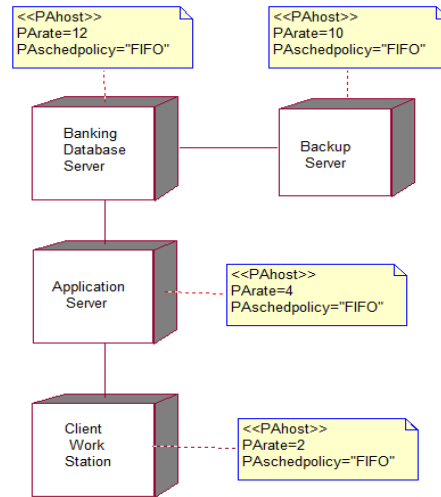
In other words, arrival rate is computed based on use case (based on PAoccurrence), which results in using sequence diagram. We add one parameter on fuzzy transition of F-CPN model to incorporate fuzzy approach in algorithm (corresponding to possession degree of fuzzy transition in fuzzy sequence diagram). Fuzzy token states the way of using sources as fuzzy).

### 5. Results and Discussion

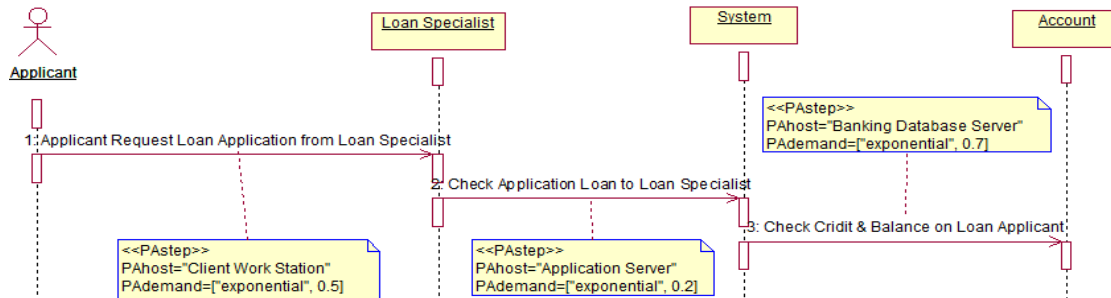
Case study of the proposed model of this paper is associated with facility section in a hypothetical bank. Fig. 4 shows diagram of use case just with one use case. Deployment diagram related to system has been shown in Fig. 5. Fig. 6 shows sequence diagram resulting in implementation of user case.



**Fig. 4.** use case diagram along with annotation performance



**Fig.5.** Deployment diagram along with Annotation Performance



**Fig. 6.** Part of Fuzzy Sequence Diagram Beside Annotation Performance

Middle graph indicating source and frequency of their using is considered in Fig. 7. In this graph, one node is considered for each message in sequence shown in Fig. 7, by numbering available sources in settlement diagram, we observe that one additional unit will be added to indicator, if one source is used again. Fig. 8 shows colored Petri net associated with middle graph. Based on the assumption that use case associated with sequence diagram has tag of `PAoccurrence=["exponential", 10]`, we will have:  $\lambda=1/10$  and `PAdemand` (for activities which use resource of application server) maintains these values, respectively. In addition, (Color Petri net has four classes):  $\{0.2, 0.3, 0.2, 0.2\}$ , fire rate in transition of Petri net for resource of Application Server equals in:

$$SR = \text{rate} [\text{res}] / \text{demand} [\text{act}] = (4/0.2, 4/0.3, 4/0.2, 4/0.2) = (20, 13.3, 20, 20)$$

By specifying arrival rate ( $\lambda$ ) and fire rate (service), we can analysis Petri net. Implementation tool is Rational Rose (ver 7) for drawing UML diagram and CPN is used for stimulating color Petri net. Computations associated with fuzzy logic have been performed by MATLAB. CPN can study meters associated with efficiency. Exploitation operational power and queue length have been emphasized in this study. Evidently, the number of available tokens in each place indicates queue length.

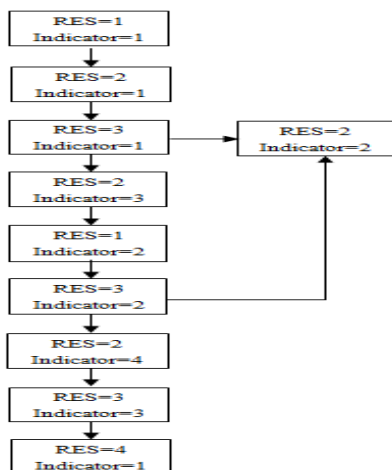


Fig.7. Pragmatic model and formal model

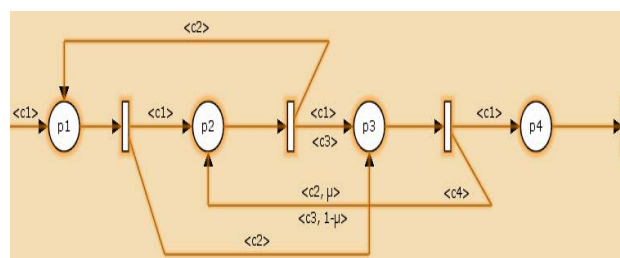


Fig.8. Model F-CPN Coordinate Fuzzy Sequence Diagram

## 6. Conclusions

In this article, a strategy was proposed for efficiency evaluation of software architecture. Providing fuzzy color Petri net based on fuzzy UML helps us consider evaluating non-functionality needs (efficiency). CPN is one of the best executive models because it combines Petri nets features with high level programming language used in this article. In addition, the implementation of fuzzy logic has improved system performance compared with the performance of the other systems.

## References

- Balsamo, S., Mamprin, R., & Marzolla, M. (2004) Performance evaluation of software architectures with queuing network models, *Proceeding of ESMc 04, Paris, France*, October 25-27.
- Emadi, S. (2008) Providing a model for studying executive capability of Software Architecture Ph.D thesis, Computer faculty of Islamic Azad University in Tehran Science and researches Unit.
- Eshuis, R., & Wieringa, R. (2000). Requirements-level semantics for UML statecharts. *Fourth International Conference on Formal methods for open object-based distributed systems IV*, Kluwer Academic Publishers, Norwell, MA, USA , pp. 121--140 .
- Gyarmati, E., & Strakendal, P. (2002). Software Performance Prediction-Using SPE, Master Thesis Software Engineering, Department of Software Engineering and Computer Science Blekinge Institute of Technology.
- Harounabadi, A. (2006). Design an intelligent model in applying fuzzy object oriented databases for systems development. Ph.D. Proposal, Science and Research Branch, Azad University, Tehran, Iran.
- Harounabadi, A. (2007) Using fuzzy UML for modeling uncertainty systems. *Seventh Conference of fuzzy systems Mashhad Ferdosi University*.
- Harounabadi A. (2010) Modeling and evaluating information systems by Fuzzy color Petri Net. *Sixteenth Annual National Conference of Iran Computer Association Tehran Sharif University*.
- Kant, K. (1992). *Introduction to Computer System Performance Evaluation*. Mc Graw-Hill, Inc.
- Kumar, B., & Jaspernete, J. (2012). UML profiles for modeling real-time communication protocols. *Journal of Object Technology*, 9(2), 178-198.
- Ma, Z.M. (2005). *Fuzzy information modeling with the UML, Advanced in Fuzzy object oriented databases: Modeling and applications*. Idea Group Publishing.
- Merseguer, J., & Campos, J. (2004). Software performance modeling using UML and Petri Nets. *Lecture Notes in Computer Science*, 2965, 265-289.
- Motament, H. (2009). Transforming fuzzy state diagram to fuzzy Petri Net. *Journal of Computer Engineering*, 1, 29-44.
- Pettit, R., & Gomaa, H. (2004). Modeling Behavioral Patterns of Concurrent Software Architectures Using Petri Nets. *Proceeding of the Fourth Working IEEE/IFIP Conference on Software Architecture*.
- Pour Kamali Anaraki, M. (2001) Improving methods of evaluating software architecture. *Amir kabir university*.
- Saldhana, J. A., & Shatz, Sol M. (2000). UML diagrams to object Petri Net models: An approach for modeling and analysis, *In International Conference on Software Engineering and Knowledge Engineering*, 103-110.