Research Journal of Applied Sciences, Engineering and Technology 6(20): 3703-3714, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 21, 2011 Accepted: February 22, 2012 Published: November 10, 2013

Modeling Sequence Diagram in Fuzzy Uml to Fuzzy Petri-Net for Calculating Reliability Parameter

[1]Reza Noorian Talouki and [2]Homayon Motameni
[1]Young Researchers Club, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran
[2]Department of Computer Engineering, Islamic Azad University, Sari Branch, Sari, Iran

Abstract: Design quality in soft-ware systems is based on to achieving the non-functional parameters. It's obvious that lack of this ability in Fuzzy UML modeling can't response the needs of customers and market. Some times in real word, problems happened in non-accurate and un-certain manners. Because of this Fuzzy UML modeling be a lionize manner for analysts. This model will help us to survey the un-certain problem. Here you see the sequence diagram in Fuzzy UML that map with Fuzzy Petri-net. Thus the ability of formal models will add to semi-formal Fuzzy UML. Formalization will add the automatic process ability to semiofficial Fuzzy UML. Automatic achieving to non-functional parameters like reliability in experimental system and survey the accuracy of done design are it's another benefits. The most important benefit of this model is reducing the cost because no experimental model will make before implementation. Usages of Fuzzy UML map to Fuzzy Petri-net, is obvious in control systems, critical and real-time.

Keywords: Formalization fuzzy UML, fuzzy petri-net, non-functional parameters, sequence diagram

INTRODUCTION

In object-oriented system, tasks done with interaction of objects with each other and exchange the massages to each other. Interaction diagrams use for object interaction modeling. Interaction is a method that does a task by sending messages between objects. UML has various types of interaction diagrams like: sequence diagram, collaboration diagram, state diagram and scheduled diagram.

As you know UML is one of the most popular modeling languages for modeling and soft-ware development. Its semi-formal feature is a defect in the first steps of soft-ware production and this defect impress in non-functional parameters predictions and verification operations. This defect is appeared in controller, critical, reaction, real-time systems.

Since most of the world's information is uncertain, analyzers use Fuzzy UML in a spread spectrum. A lot of studies are done to eliminate this defect. Some of these studies only use mapping algorithm. This method convert UML model to a Petri-net as a math and formal model that has a visual aspect of modeling and do verification operation with a higher speed (Faul, 2004; Shin *et al*., 2003; Bernardi *et al*., 2002; Eshuis, 2002; Pettit and Gomaa, 2002; Saldhana and Shatz, 2000; Elkoutbi and Rodulf Keller, 1998; Bernardinello and De Cindio, 1992).

Some of these algorithms not only use mapping algorithms but also evaluate the efficiency of non-functional and parameters in Petri-net that produced by UML model (Balsamo *et al*., 2004; Merseguer *et al*., 2002, 2004; Fukuzawa and Saeki, 2002).

In the previous studies in addition to providing mapping models for some common types of UML diagrams, especially state and activity diagrams, they were provided with some methods for evaluating quality parameters (Motameni, 2006; Motameni *et al*., 2006a, b, 2005a, b, 2008a, b).

It is obvious that lack of the abilities like compute non-functional parameters in UML, never ingratiate customers and market needs. recent studies not only mapping algorithms for some of popular UML diagrams, specially activity and state diagram but also present methods to evaluate some of quality parameters.

This study presents an algorithm for mapping sequence diagram in Fuzzy UML to Fuzzy Petri-net. This algorithm can help us to increase the Reliability ability of the system before implementation. This feature can help us to reduce the cost. Moreover this feature adds Formalization ability and automatic processing to Fuzzy UML model.

First we introduce Fuzzy sequence diagram and Fuzzy Petri-net and then we describe an algorithm for mapping created sequence diagram in Fuzzy UML to

Corresponding Author: Reza Noorian Talouki, Young Researchers Club, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran

fuzzy Petri-net and compute Reliability ability for Fuzzy Petri-net. In the end we survey a laundry case-study.

FUZZY UML

Nowadays UML is an important instrument in object-oriented system. This language has visual modeling ability. This ability enable manufactures by systems, increase views in a standard level in a presented design and be an effective mechanism in communication with another designs.

Object Management Group (2005) and Rumbuaugh *et al*. (2005) because real words information are often un-certain, so in many cases we can't modeling such information with UML. To solve this problem Fuzzy UML be created. With using Fuzzy UML, significance can model un-certain significance (Wang, 2005; Zongmin, 2005; Ma, 1999).

Fuzzy sequence diagram: In UML we use sequence diagram for realizing use-case. If the highlighted use-case be un-certain, the sequence diagram will be un-certain.to indicate the uncertain massages on mentioned diagram we use a vertical line at the first of the message. Figure 1 shows this feature.

Some of the messages in sequence diagram can be converted to methods. Quatrain (1999) Un-certainty in method has two-level of fuzzification. 1^{st} method belongs you with dependency grade. 2^{nd} nature of the method is fuzzy. Figure 1 shows the C method that produced from C message. And this belongs to B object with dependency grade between 0 and 1(first step of fuzzification). But D message and follow it D method have un-certain nature (2^{nd} step of fuzzification). Dependency grade in a sequence diagram and dependency grade of use-case both will show with$(T.norm)$. For example in Fig. 1 we have:

$$T[\mu_B(x), \mu_D(x)] = \mu_{B\cap D}(x)$$
$$where\ t: [0,1] \times [0,1] \rightarrow [0,1]$$

In soft-ware system we encounter to scenarios that designed them are so criticize. The fuzzy scenarios that proposed in Fuzzy-UML present a solution for this. We can convert the scenarios to fuzzy composite rules with finding the effective parameters of them. Figure 2 shows a sequence diagram that extracts from supposition scenario. In fuzzy aspect to assistance the fuzzy scenarios we use Fuzzy membership rule.

Fuzzy membership rule is a rule that describe the relation between two transitions**.** If the first section of Fuzzy membership rule be include of (AND and OR) connectors, it will be a composite Fuzzy membership rule. Opposite of Fig. 2 that shows a crisp aspect Fig. 3 shows a fuzzy method. Programming language do the implementation. Figure 4 shows the sample of done job in java language for sequence diagram.

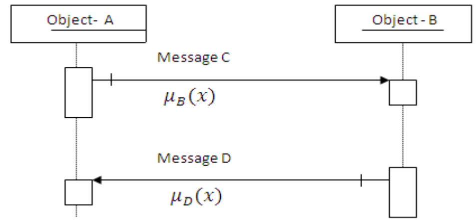

Fig. 1: Fuzzy message in Fuzzy sequence diagram

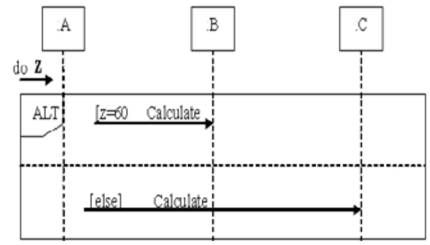

Fig. 2: Crisp sequence diagram

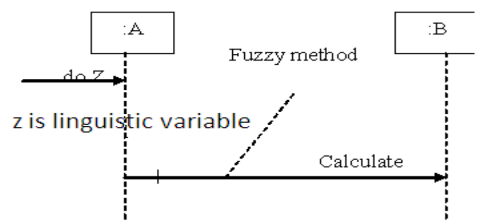

Fig. 3: Fuzzy Sequence diagram for a sample scenario

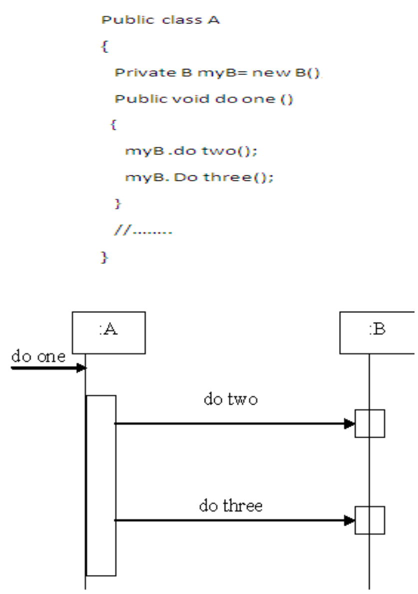

Fig. 4: An example of the code created for sequence diagrams

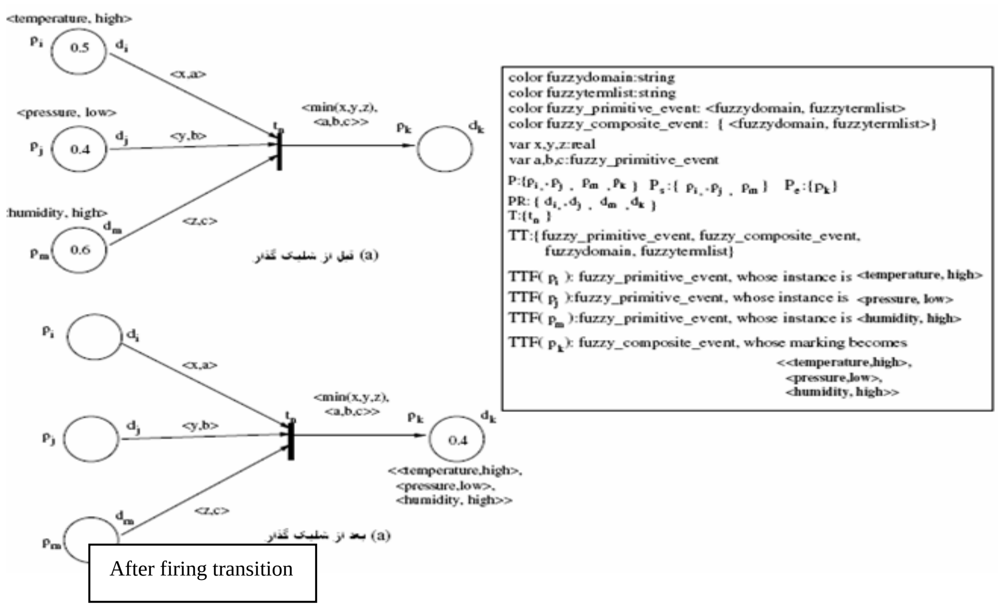

Fig. 5: Firing a fuzzy Petri net

Fuzzy Petri-net: Fuzzy Petri-net used for modeling the fuzzy rules (Fig. 5). Not only have the abilities like concurrency, visually, formalization, math-based and etceterabut also has the ability for using fuzzy variable. Fuzzy petrinet has the following abilities: (Bostan-Korpeoglu and Yazici, 2006)

- Fuzzy values can place in locations, transitions and signs
- It can transfer the parameters.it means that we can transfer the parameter from a location to another location
- we can use events and conditions for modeling the rules.it means that we can use the events and conditions to create the rules
- we can use the various kind of active and apriori rules for modeling
- it has the fuzzy deduction
- Fuzzy Petri-net gives us ability to Simulation of concurrent issues
- We can use it to analyze the static rules

The structure of a fuzzy Petri-net has shown in the follow equation:

$$(P, P_S, P_\epsilon, T, T_F, TRTF, A, I, O, TT, TTF, AEF, PR, PPM, TV)$$

- P is a limited collection of fuzzy places, that each place has its own special features
- $P_S \subset P$ Is a limited collection of input places for primary events
- $P_\epsilon \subset P$ Is a limited collection of output places for operations or results
- T is a limited collection of Fuzzy transition, that produce the values for output places based on the provided values by input places
- TF Is a limited collection of Fuzzy transition and do the fuzzy inference operations
- $TRTF: T \rightarrow T_F$ Is a transition function and map the each transaction that is a member of T to a TF function
- A Is a limited collection of arcs**,** to connect the places and transitions.connectors between input places & transitions $(P \times T)$ and between output places & transitions $(\mathrm{T} \times P)$ created by arcs
- $A \subseteq (P \times T \cup \mathrm{T} \times P)$
- $I: P \rightarrow T$ is an input map
- $O: T \rightarrow P$ is an output map
- TT is a limited collection of fuzzy token types. Each token has its own value that is assign by a membership function
- TTF is a limited collection of token types and map each fuzzy place that is a member of P to fuzzy token types that is a member of (TT):
- $AEF : Arc \rightarrow Expression$ Is an Expression-function on an arc and map each arc to an expression that is consisting of information's
- PR is a is a limited collection of transitions and it I match with events, conditions, operations and results
- $PPM: P \rightarrow PR$ map a fu
- zzy place to transition
- $TV: P \rightarrow [0,1]$ is a real value of tokens and it is consist of informations about membership grade of each token to a special place

As an example we have modeling sample of fuzzy active rule:

- $(IF\ d_i\ AND\ d_j\ AND\ d_m\ THEN\ d_k)$
- $PPM(P_m) = d_m, PPM(P_j) = d_j, PPM(P_i) = d_i$
- $TV(P_i) = \mu_i = 0.5, PPM(P_k) = d_k$
- $TV(P_j) = \mu_j = 0.4$
- $TV(P_m) = \mu_m = 0.6$

If $(\mu_m > 0, \mu_j > 0, \mu_k > 0)$ then t_n transition will active and fire.

Tokens cut from $I(t_n)$, it is consist of pi، pj، pk and add to $O\ (t_n)$, that is consist of p_k. Suppose that $TF(t_n) = TRTF(t_n)$ transition function be a min operator then value of output token (membership grade) compute as follow:

$$TV(P_k) = TF_n(I(t_n)) = \min(\mu_i, \mu_j, \mu_m) = 0.4$$

Mapping algorithm of sequence diagram in Fuzzy UML to Fuzzy Petri-net and its reliability: Here we present an algorithm for mapping the sequence diagram in fuzzy UML to Fuzzy Petri-net and its reliability. This algorithm used for formalize the semiofficial model of Fuzzy UML.

First step:

compute the fuzzy value for each of variables with using membership function: We should recognize the Conditions and system situation after running for each message and also its membership function for all crisp variable. Figure 6 shows a created rules consist of conditions and system situation after running condition as a sample.

And also we present a sample of membership function for a special event (variable) in Fig. 7 and 8. It is obvious that membership function for each of language variable designed specialist.

To complete this step, for each of events, we compute all possible states for each of language variable from membership function and assign a location for each state (the first rule of fuzzification).

Rule	Event	Condition	State
R1	e1 is e11	C1	S1
R2	e1 is e11 e2 is e21	C1 AND C2	S2
R3	e1 is e13	C3	S3

C1 : IF (e1 is e11) THEN ...
C2 : IF (e2 is e21) THEN ...
C3 : IF (e1 is e13) THEN ...

Fig. 6: An example of created rules

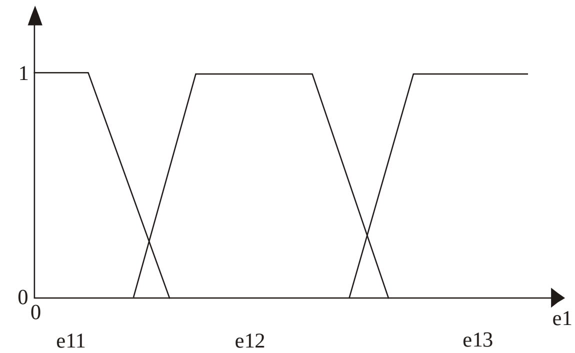

Fig. 7: Membership function for the event (variable) e1

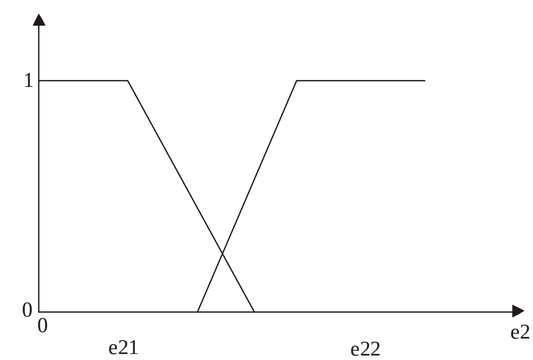

Fig. 8: Membership function for the event (variable) e_2

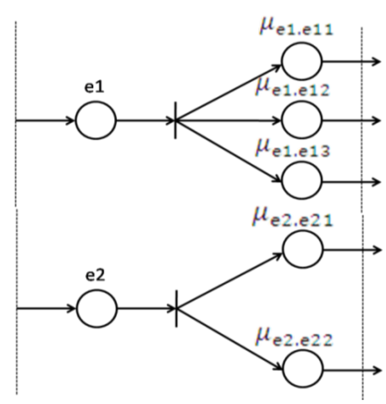

Fig. 9: Calculate the fuzzy value for each variable using membership functions

As an example, for membership function, e_1, we compute the first step of Petri-net. You can see it in Fig. 9.

2 nd Step: compute the output value of rules in fuzzy manner: Petri-net making from fuzzy rules compute in first step and also fuzzy events, checking conditions (compute the result of rules) and transfer the results to the next place are compute in first step.

First we should check the accuracy of Rules Between [0, 1]. So we should create a map for it. For each rules we create a transition to check the accuracy.

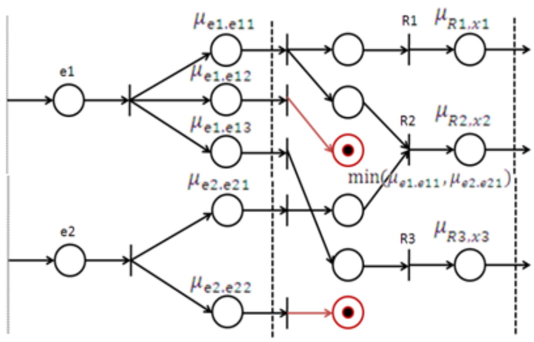

Fig. 10: The second step, calculate the amount of output fuzzy rules

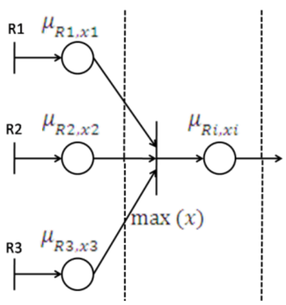

Fig. 11: Extracting a fuzzy rule and its value, from the fuzzy rules

The results transfer to the next place of net. It means that, token continues its life cycle with the given fuzzy value. Token can have a value batween 0 and 1. The results at end of C condition analysis are equel to the primary value of fuzzy value.

In this section we connect the places with the values of first step to the considered place, for each rule. Figure 10 show this procedure. we also connect the useless transition to the end places.

3rd Step:combination of fuzzy rules: In this section we have a place for each rule. Here, each fuzzy token has two part: rule name and fuzzy value.we fire the token with the most fuzzy value with its rule number to the next place. This show the rule that its event has the most fuzzy state. Figure 11 shows the extraction of a rule and its fuzzy value with using the presented transition.

4th Step: add fuzzy state of method to the Fuzzy Petri-net: In the previous steps we check the fuzzy state for method's input. It means that heretofore input of methods was fuzzy (1st level of fuzzification).

But if the method has fuzzy state, it will check in this step. We add the fuzzy state of method to fuzzy petri-net. The result is a combination of fuzzy state of the rules (1st step of fuzzification) and fuzzy state of methods (2nd level of fuzzification). It compute as below:

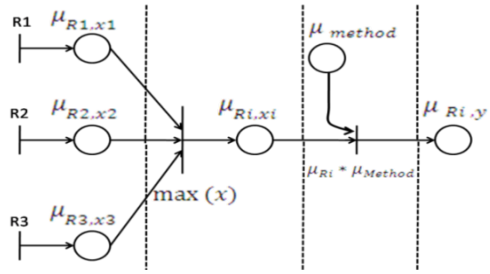

Fig. 12: Add fuzzy mode, to the method of running

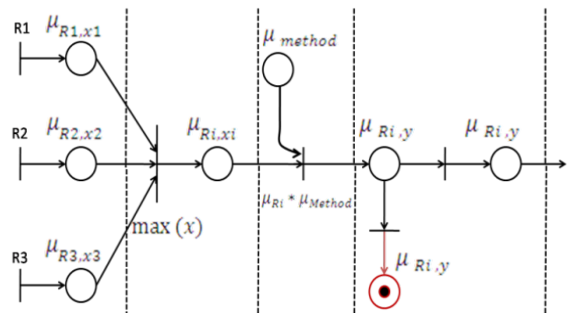

Fig. 13: Continue or end the life of tokens

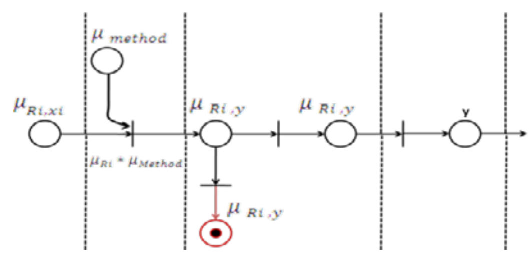

Fig. 14: Petri net defuzzification

$$\mu_{Ri,h} = (\mu_{Ri,xi} * \mu_{method})$$

It is obvious that the fuzzy value of the event can be a value between 0 and 1.

$$\mu_{method} = [0,1]$$

If the method doesn't have fuzzy state we will ignore this step. Or we can allocate 1 value instead of fuzzy value of method.

$$\mu_{method} = 1$$

Figure 12 shows the added 2nd fuzzy level to exist net by a membership function.

5th Step: survey the end or continuous of token's life in Fuzzy Petri-net: After adding the fuzzy state of a method to the net, maybe fuzzy variable get an invalid value that don't have the permission for staying on net. Because of this we should emit this token from fuzzy petri-net. To show the emitted token we should show the end place. Figure 13 shows, whether a token stay in life cycle or not.

6th Step: defuzzification and compute the total value of fuzzy Petri-net: Same in the first and middle steps, that fuzzification help us to evaluate the rules, the total result also should be a certain result after the combination of output rules. Because of this defuzzification should be done till a certain value achieved from fuzzy collection instead of a collection of values.

There are various methods for defuzzification, Like COG (Center of Gravity) method, COA (Center of Area) method, COS (Center of Sum) method and MOM (Mean of Maxima) method. We use center of gravity method here. This shows in the below equation:

$$COG = \int_a^b \mu_{A(x)}\, x\, dx \Big/ \int_a^b \mu_{A(x)}\, dx$$

To measuring the sum we use the last fuzzy value. Token will be used for applying the best previous value to resultant of membership value. Doing this act will shown in Fig. 14.

Result of this step is a combination of rules result with each other, like Fig. 7. Fuzzy collection is a combination of outputs. We should use defuzzification operation, because the total output of fuzzy system should be a certain value. Input of defuzzification process is a fuzzy collection and output of it is a crisp value.

7th Step: compute reliability for Fuzzy Petri-net: To compute the reliability of a system we should define a success rate (D) for token. Success rate assign the probability of fired tokens, if there is no problem. In another word $(1 - D)$ is probability of missing the data's. The value that used for computing the success rate of next place is a token that there is in the input of $(t1)$. Token. Success rate changes to $(D * F)$ when token be fired.

For example after firing the $(t1)$ token, success rate of token changes from (D) to $(D * F)$. Also we can assign the reliability rate of net equal to success rate. And compute the Non-Functional factor of reliability for Petri-net. Figure 15 shows the compute method of reliability in a supposition net.

After designing the fuzzy Petri-net work, system miss its semiofficial feature and convert to a formal system. After formalization of sequence diagram some abilities add to system, like, compute reliability for sequence diagram automatically. To compute reliability we need some values. First we should compute the reliability of each rule and each of language variables. Also we need the reliability of method's fuzzy value (μ_{Method}).

We need an algorithm to compute the reliability of sequence diagram automatically. We present an algorithm to this study:

- We assign the reliability of places that are input variable equal to reliability of each input variable
- We assign the reliability of tokens that considered for each rule equal to reliability of rule
- We set the reliability of method's fuzzy state (μ_{Method}) in the considered place for method
- We set the reliability of other Transitions equal to And compute the reliability of system's output

The benefits of presented algorithm: The benefits of this algorithm list as follow:

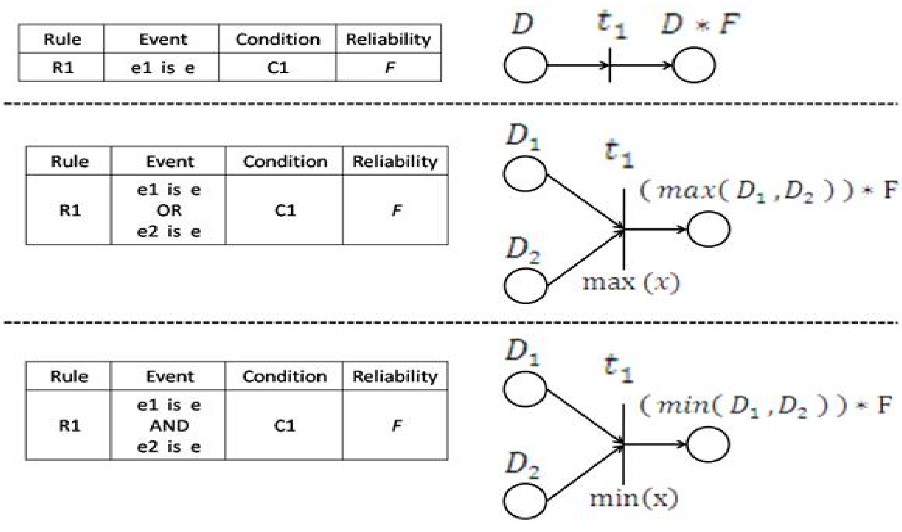

Fig. 15: Calculate reliability using Petri net

- Adding the un-crispy in modeling informational systems
- Using a standard language and applying its benefits to compatibility the different parts of soft-ware extension
- Ability of running this method on complex systems
- Formalization the semiofficial model of Fuzzy UML for using the benefits of formal models
- Automatism mapping sequence diagram in Fuzzy UML to fuzzy Petri-net
- Adding the ability of computation the non-functional factors of an under-constructed system, before implementation and in system's analyze phase
- Reduced the cost by compute the non-functional factors of an under-constructed system without making primary sample (non-functional factors measured in analyze phase)

Example:

Creating a Fuzzy Petri net for sequence diagram of a washing machine: We want to design a Petri Fuzzy net for sequence diagram of a washing machine. This system should consider the extend of dirty and greasy of clothes and then determine theme of washing. Then call a method for adding redolent, this method run as a fuzzy method (2nd level of fuzzy). For example rate of cloth's ambrosia in fuzzy method is 0.75.

Figure 16 shows the sequence diagram of a washing machine that, its inputs are fuzzy variables and language variables.

To run this task first we should recognize all rules, conditions and crispy fuzzy events. Figure 17 shows the rules of first method.

Also we should design membership function for each of fuzzy event. It is obvious that membership function designment should be done by a specialist. Figure 18 shows the antecedent and consequent of 1th method.

Inputs of first method were fuzzy but the method was not. Then a fuzzy method call and the fuzzy variable, redolent, will be added to the system. Hereby the quantity of aroma be recognized. Figure 19 shows the rule of 2nd method. In this method input variables of method and method both are fuzzy.

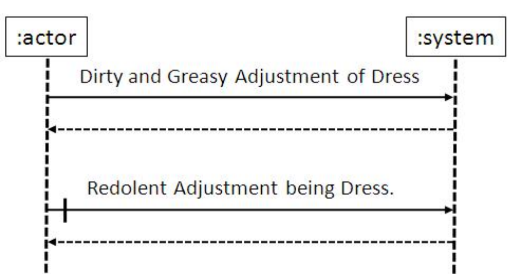

Fig. 16: Sequence diagram of a washing machine

Rule	Event	Condition	State
R1	Dress is Dirty	IF (Dress is Much Dirty) then	Wash time is Min
R2	Dress is Greasy	IF (Dress is Little Greasy) then	Wash time is Medium
R3	Dress is Dirty And Dress is Greasy	IF (Dress is Much Dirty) And IF (Dress is Much Greasy) then	Wash time is Long

Fig. 17: Rules of first method

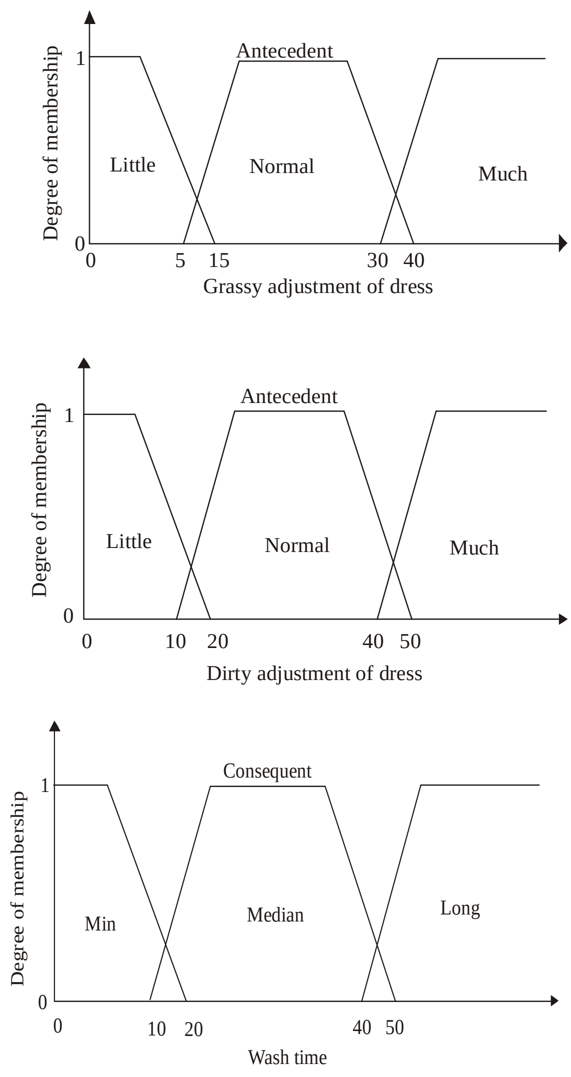

Fig. 18: Showing Membership functions of fuzzy variables

Rule	Event	Condition	State
R1	Dress is Redolent	IF (Mach Redolent Require) then	Amount Of Redolent Stuff is Many
R2	Dress is Redolent	IF (Little Redolent Require) then	Amount Of Redolent Stuff is Low

Fig. 19: Rules of 2nd method

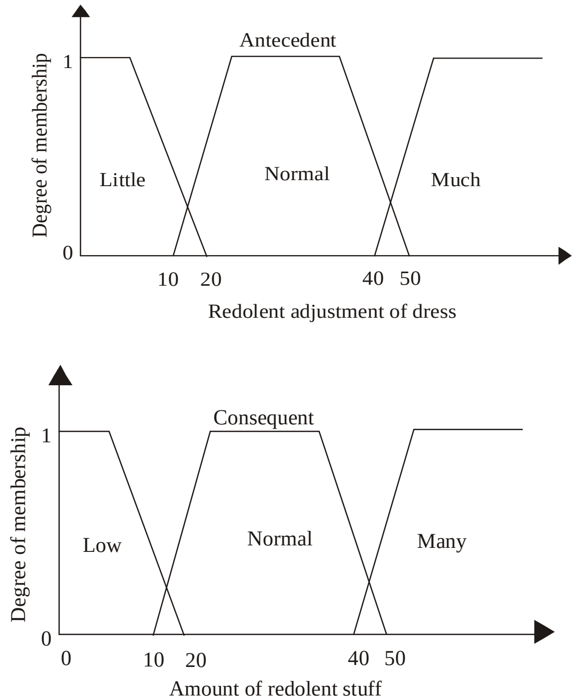

Fig. 20: Showing Membership functions of fuzzy variables in 2nd method

Also we should design the membership function for each fuzzy event of 2nd method. Figure 20 shows the antecedent and consequent of 2nd method.

Now we have all rules and membership function and we can design the sequence diagram of washing machine, with using the previous described steps. You can see it in Fig. 21.

Calculate the output of Petri-net that produced by sequence diagram: Now we show how compute the output of Fuzzy Petri-net that produced by sequence diagram. Our variables should be fuzzy variables.

Example: consider dirty of clothe is so much (45), greasy of clothe is much (40) and happening the method is crisp ($\mu_{Method} = 1$). We wish, our net, recognize the best washing time based on input variables:

$$\mu_{Dirty\ .Much} = 0.5$$
$$\mu_{Dirty\ .Normal} = 0.5$$
$$\mu_{Dirty\ .Little} = 0$$
$$\mu_{Greasy\ .Much} = 1$$
$$\mu_{Greasy\ .Normal} = 0$$
$$\mu_{Greasy\ .Little} = 0$$

Now, for 2nd method we should receive fuzzy variables.

For example: redolence of clothe is little to normal (15):

$$\mu_{Redolent\ .Much} = 0$$
$$\mu_{Redolent\ .Normal} = 0.5$$
$$\mu_{Redolent\ .Little} = 0.5$$

Here the method has a fuzzy value. For example (0.75), this means redolenting task (method) is done very good:

$$\mu_{Method} = 0.75$$

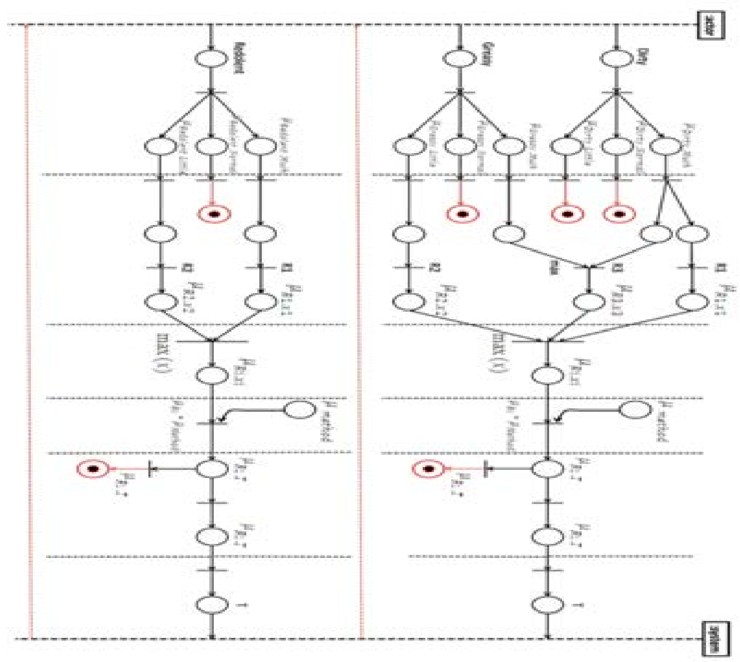

Fig. 21: Sequence diagram of washing machine

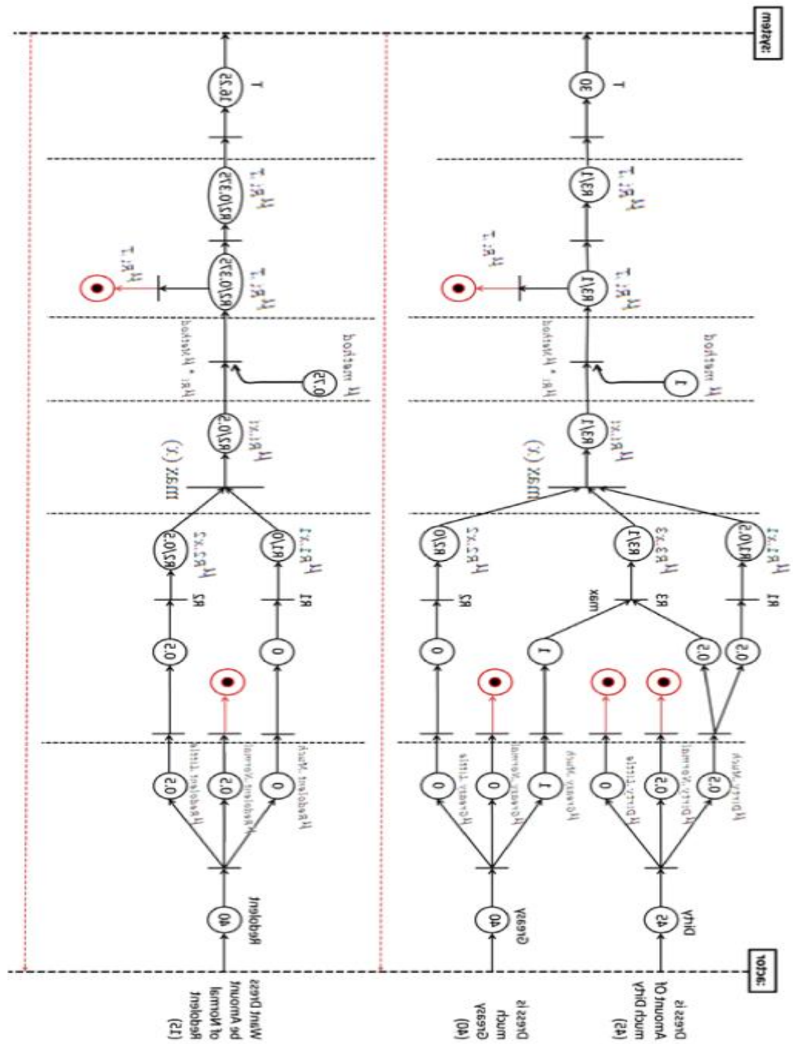

Fig. 22: Result achieved, after running Petri-net for special fuzzy value

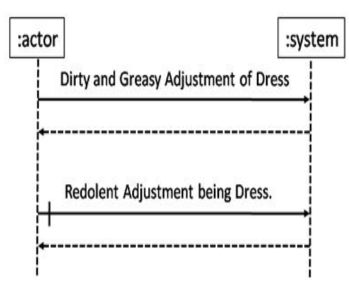

Fig. 23: Fuzzy sequence diagram, modeled by fuzzy Petri-net for washing machine

You see that we can achieve the certain computed value from fuzzy value by adding fuzzy value to system and use it. Below result achieved, after running this Petri-net for special fuzzy value (Fig. 22).

Compute the reliability of Fuzzy Petri-net for washing machine: Figure 23 shows the fuzzy sequence diagram, modeled by Fuzzy Petri-net. The above algorithm used for computing each of input messages or actor-system messages.

Rule	Event	Condition	State	Reliability
R1	Dress is Dirty	IF (Dress is Much Dirty) then	Wash time is Min	1
R2	Dress is Greasy	IF (Dress is Little Greasy) then	Wash time is Medium	1
R3	Dress is Dirty And Dress is Greasy	IF (Dress is Much Dirty) And IF (Dress is Much Greasy) then	Wash time is Long	0.95

Reliability Of Dirtiness = 0.95

Reliability Of Greasiness = 0.95

$$\mu_{Method_1} = 1$$

Rule	Event	Condition	State	Reliability
R1	Dress is Redolent	IF (Mach Redolent Require) then	Amount Of Redolent Stuff is Many	1
R2	Dress is Redolent	IF (Little Redolent Require) then	Amount Of Redolent Stuff is Low	1

Reliability Of Redolence = 0.95

$$\mu_{Method_2} = 0.75$$

Fig. 24: Adding reliability to the existing rules

Figure 24 has shown the reliability of each rules and their fuzzy variable and reliability of method's fuzzy value for washing machine.

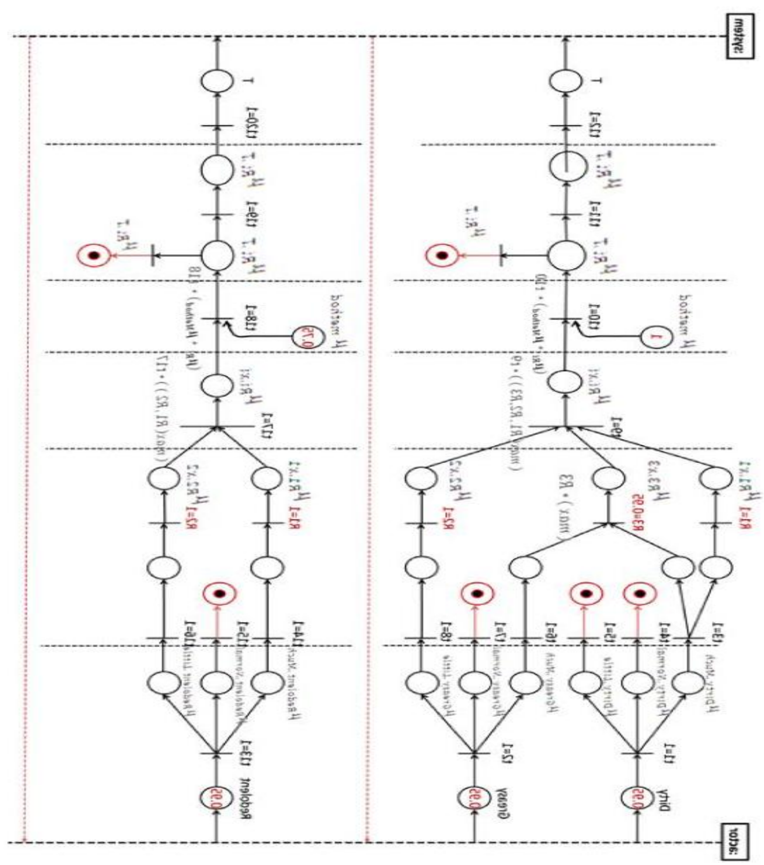

Fig. 25: Inserting primary value of tokens for computing the reliability in Petri-net

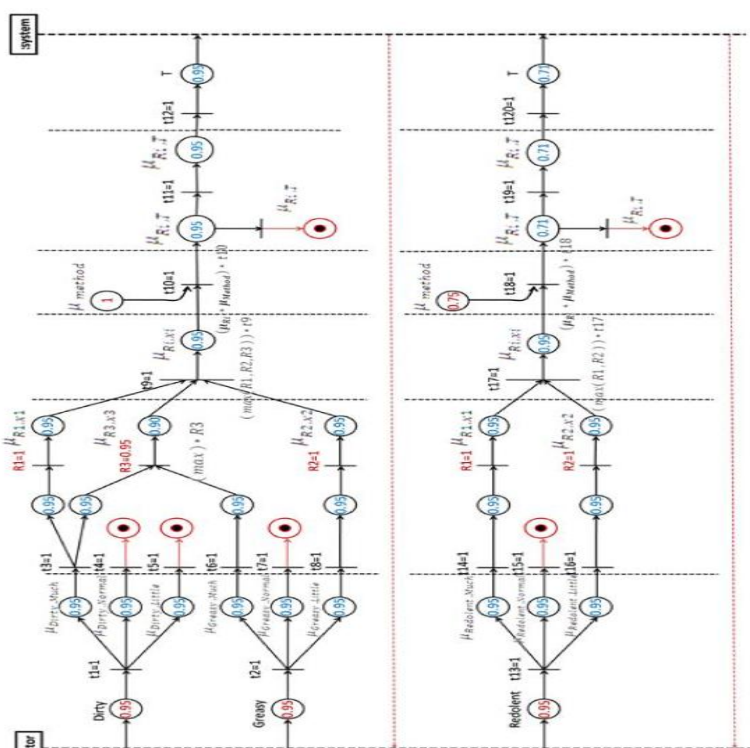

Fig. 26: Computing the reliability in Petri-net

Now we can achieve the reliability for each of sequence diagram's method with allocate the above value. Figure 25 has shown the primary value of Tokens for computing the reliability in Petri-net.

Figure 26 shows the steps of reliability computing and its total result.

Reliability value of first method is 0.95. This ability is a high ability for fuzzy methods. But 2nd method, with 0.71 Reliability value has low reliability. System analyzer should run this method with a better way. Method's fuzzy feature is the main reason for reliability.

System's developer should created systems with more reliability. In another word tokens should have more successfully rate.

RUP emit this problem, during sequence repetition. When we consider designing a controller system or a critical one, Reliability shows its necessity. Reliability is the one of the important factors in soft-ware engineering. Analyzers shouldn't ignore this factor.

CONCLUSION

Convert the semi-official model of Fuzzy UML to formal Fuzzy Petri-net, present the abilities like formulation, graphical sign, automatic process and task analyzing.

Recently, making a primary sample was a necessity, Because of need to computing the non-functional parameters like reliability and tolerance and etcetera, before system implementation. We try to Automatism mapping sequence diagram in Fuzzy UML to fuzzy Petri-net.

Our approach is computing this parameter before making primary sample and in automatic manner. This automation reduces the cost of software making. This algorithm spread the software engineer's ability to guess the nonlinear problem and never force them to making a mathematical model for our world. Lack of the math-metical models using is the most important feature of this algorithm. It means that, this algorithm estimate and evaluates the model, without using math-metical models.

Also this algorithm can help us to formalize the sequence diagram in complex systems. Formalization can compute non-functional factors without making primary sample.

For future works we can survey other diagrams to extract non-functional parameters. For extract non-functional parameters we can use mathematic formulas or we can convert Fuzzy UML to a formal model with help of Fuzzy Petri-net. Then we can extract the non-functional parameters.

REFERENCES

Balsamo, S., A. Di Marco, P. Inverardi and M. Simeoni, 2004. Model-based performance prediction in software development: A survey. IEEE Trans. Software Eng., 30(5): 295-310.

Bernardinello, L. and F. De Cindio, 1992. A Survey of Basic Net Models and Modular Net Classes. LNCS, Springer-Verlag, pp: 609.

Bernardi, S., S. Donatelli and J. Merseguer, 2002. From UML sequence diagrams and statecharts to analysable petri net models. ACM Proceeding Int'l Workshop Software and Performance.

Bostan-Korpeoglu, B. and A. Yazici, 2006. A Fuzzy Petri Net Model for Intelligent Databases. Department of Computer Engineering, Middle East Technical University, Elsevier Ankara.

Elkoutbi, M. and K. Rodulf Keller, 1998. Modeling interactive systems with hierarchical colored petri nets. Advanced Simulation Technologies Confecnere Boston, pp: 432-437.

Eshuis, R., 2002. Semantics and verification of UML activity diagrams for workflow modelling. Ph.D. Thesis, University of Twente.

Faul, B.M., 2004. Verifiable modeling techniques using a colored Petri net graphical language. Technol. Rev. J., Spring/Summer.

Fukuzawa, K. and M. Saeki, 2002. Evaluating software architecture by colored petri net. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE), pp: 263-270.

Ma, Z.M., 1999. Extending UML for Fuzzy information modeling in Object_Oriented Database. Theories and Practices, Idea Group Publishing.

Merseguer, J., S. Bernardi, J. Campos and S. Donatelli, 2002. A compositional semantics for UML state machines aimed at performance evaluation. Proceeding of the 6th International Workshop on Discrete Event Systems (WODES), Zaragoza, Spain, pp: 295-302.

Merseguer, J., J.P. LopezGrao and J. Campos, 2004. From UML Activity Diagrams to Stochastic Petri Nets: Application to Software Performance Engineering. ACM, WOSP 04 January.

Motameni, H., M. Mozafari and A. Movaghar, 2005a. Evaluating UML State Diagrams using Colored Petri Net. SYNASC.

Motameni, H., *et al*., 2005b. Verifying and evaluating UML activity diagram by converting to CPN. Proceeding of SYNASC, Romania.

Motameni, H., 2006. Using markov theory for deriving non-functional parameters on transformed petri net from activity diagram. Proceeding of Software Engineering Conference, Russia.

Motameni, H., A. Movaghar, M. Siasifar, M. Zandakbari and H. Montazeri, 2006a. Mapping state diagram to petri net: An approach tousemarkov theory for analyzing non-functional parameters. IEEE International Conferenceon Computer, Information and System Science, University of Bridgport.

Motameni, H., M. Zandakbari and Movaghar, 2006b. Deriving performance parameters from the activity diagram using gspn and Markov Chain. Proceeding of 4th International Conference on Computer Science and its Aapplications, (ICCSA), San Ddiego,California.

Motameni, H., A. Movaghar, I. Daneshfar, H. Nemat Zadeh and J. Bakhshi, 2008. Mapping to convert activity diagram in fuzzy UML to fuzzy petri net. World Appl. Sci. J., 3(3): 514-521.

Object Management Group, 2005. UMLTM Profile for Schedulability. Performance and Time Specification, OMG Document, Version 1.1.

Pettit, R.G. and H. Gomaa, 2002. Validation of Dynamic Behavior in UML using Colored Petri Nets. UML, Zaragoza, Spain, pp: 295-302.

Quatrain, T., 1999. Visual Modeling with Rational Rose 2000 and UML. Addison Wesley, Reading, Mass.

Rumbuaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, 2005. Object-Oreinted Modeling and Design. Prentice Hall, Englewood Cliffs, USA.

Saldhana, J. and S.M. Shatz, 2000. UML diagrams to object petri net models: An approach for modeling and analysis. Proceeding of the International Conference on Software Eng. and Knowledge Eng. (SEKE), Chicago, pp: 10-103.

Shin, M., A. Levis and L. Wagenhals, 2003. Transformation of UML-based system model into CPN model for validating system behavior. Proceeding of Compositional Verification of UML Models Workshop of the UML Conference, California.

Wang, L., 2005. Fuzzy UML. Seminararbeit, Sommersemester.

Zongmin, M., 2005. Fuzzy Information Modeling With the Uml, Idea.