

## *Mapping Sequence diagram in Fuzzy UML to Fuzzy Petri Net*

E.Akbari<sup>1</sup>, R. Noorian Talooki<sup>2</sup>, H. Motameni<sup>\*2</sup>

<sup>1</sup>.Department of Mathematics, Islamic Azad University, Sari Branch,  
Sari, Iran

<sup>2</sup>.Department of Computer Engineering, Islamic Azad University, Sari Branch,  
Sari, Iran

---

\*Correspondence E-mail: H. Motameni ,motameni@iausari.ac.ir

© 2013 Copyright by Islamic Azad University, Rasht Branch, Rasht, Iran

Online version is available on: [www.ijo.iaurasht.ac.ir](http://www.ijo.iaurasht.ac.ir)

---

### **Abstract**

this ability in fuzzy UML, practically leaves the customers and market's need without response in this important and vital area. Here, the available sequence diagrams in fuzzy UML will map into fuzzy Petri net. However, the formal models ability will be added to the Semi-formal fuzzy UML. This formalization will add the automatic processing ability to the Semi-formal fuzzy UML.

Further more, the other advantages of this mapping is: access to non-functional parameters such as reliability automatically to the considering systems, study the verification of the designed plan and also decrease the expenses because of satiety to make lab sample before its implementation. Using the fuzzy UML mapping into fuzzy Petri net in control, critical and real-time systems will be more applicable.

**Keywords:** non-functional parameters, fuzzy UML, sequence diagram, fuzzy Petri net, Formalization.

---

### **1.Introduction**

In object oriented systems, function tasks are done through the interaction of objects, which exchange messages with each other. Interaction diagrams are used for modeling object interactions with one another. Interaction is the method of exchanging messages with each other to do a task. In UML, there are different

forms of interactive diagrams: sequence diagrams, graphs browsing interactive charts and graphs scheduled communication.

Class diagrams, modeling the static structure of system, while the interactive diagrams, modeling the dynamic aspects of system. That means these figures show how an object is interacting with another object to accomplish the high level tasks which each of the objects alone are not able to do. Sequence diagrams are used to show this interaction, and thus messages are emphasized over time. Here, we will talk about sequence diagram.

As you know, UML\* is one of the most commonly used modeling languages for modeling and software development.

However, the semi-formal characteristic of this method is a limitation for verification operations and predicting non-functional parameters of the software, especially in the first cycle of the software production. This problem is more critical for control, critical, reactive and real time systems.

On the other hand, since the majority of the real world information is uncertain, therefore fuzzy UML diagrams have been extensively used by system analyzers. Several researches have been performed to tackle with the semi-formal problem of UML. Some of these researches have only used a transformation algorithm, which transforms the created UML model into a Petri net as a mathematical and formal model that, in turn, contains the visual aspect of modeling and pursues the verification operations with further ability [1, 2, 3, 4, 5, 6, 7, and 8].

Some of the researches in this field besides representing a transformation algorithm (or without representing an algorithm and only by using the available Algorithm); evaluate the capability of the non-operational parameters and commonly qualitative parameters on the obtained Petri nets created by the UML model [9, 10, 11, 12].

Certainly, the lack of features such as non-operational parameters in calculated UML model leaves the customer requirements and market practice without the same response in this important and sensitive areas and this problem reduces the importance and the value of such researches. In the previous studies [13, 14, 15, 16, 17] 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.

In this paper, Fuzzy UML sequence diagram is mapping to fuzzy Petri net, that this mapping added formalization and automated processing to Fuzzy UML model. This mapping will also be an introduction to achieve non-functional parameters such as reliability. At first, we will introduce Sequence charts and Fuzzy Petri networks and then we will describe an algorithm for mapping

---

\* - Unified Modeling Language

sequence diagram created in the Fuzzy UML standards, in the end as a case study, we will study the application of this model for Washing machine system.

## 2. Fuzzy UML

Now days, UML is known as one of the most important tools in extending object oriented systems. This language makes the visual modeling possible so that the system developers will be able to standardize and make the ideas understandable and establish a more effective mechanism in relations with other patterns [18, 19]. Since the real world information is mostly uncertain in a proposed general pattern, in many cases these types of information cannot be modeled by UML. Recently a model named fuzzy UML has been introduced [20, 21, 22] which has the UML characteristics, and it is also able to model uncertain concepts.

fuzzy sequence diagram:

In UML for materializing the practical cases we use the sequence diagram.

If the case is uncertain, the sequence diagram will be uncertain, too.

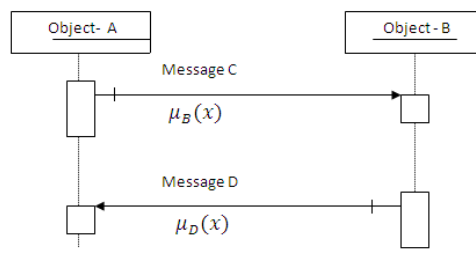


figure 1: Fuzzy method in Fuzzy sequence diagram

Most of messages in sequence diagrams are conversable to method [23].

Uncertainty in the method have two-level of fuzzification. First, the degree method belongs to you and then the second method is the logic. figure 1 C method that is rooted Post C. Degree between 0 and 1 belong to the object B is owned (the first level fuzzification). The other message D and the subsequent method D is an Uncertainty nature (the second level fuzzification). Degrees awarded methods in a sequence diagram, the user relevant degrees awarded shows. For example, in figure 1 have.

$$T[\mu_B(x), \mu_D(x)] = \mu_{B \cap D}(x)$$

where  $t : [0,1] \times [0,1] \rightarrow [0,1]$

Fuzzy Petri net:

Fuzzy Petri net are used for modeling fuzzy rules. In addition to features such as concurrency, being visual, formal being, basic math, etc, the ability to use fuzzy variables has also been added.

Fuzzy Petri net offers following capabilities [24].

- I. Permitting the implement part of the graph.
- II. Fuzzy values can be put in places, transitions and tokens.
- III. having the capability to transfer parameters (variables). The parameters of a place can be transferred to another place.
- IV. In Petri events and conditions can be used for modelling rules. (For create rules Can use events and condition.)
- V. Using type of induction and active rules for modelling.
- VI. Having fuzzy deduction(inference).
- VII. Fuzzy petri net can be used for Fuzzification, Combination, Concurrency, Composition and Sup–Min.
- VIII. It can be used to analyze the static rules.

The structure of a fuzzy Petri net is as following:

$(P, P_s, P_e, T, T_f, TRTF, A, I, O, TT, TTF, AEF, PR, PPM, TV)$

Mapping algorithm:

In this section, an algorithm for mapping the fuzzy UML sequence diagram to Fuzzy Petri net will be provided.

Step One: Calculate fuzzy value for each variable using the membership function

For each method, the condition and state of the system after the implementation (running) of method and its membership function for all variables (fuzzy / crisp) must be detects. In figure 2 an instance of the rules including the condition and status of the implementation of the system condition is presented as a sample.

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 ...

figure 2: An example of created rules

And also an example of membership function for a sample event (variable) will be provided in figure 3 and figure 4. Obviously, the membership function for each of the linguistic variables is designed by an expert.

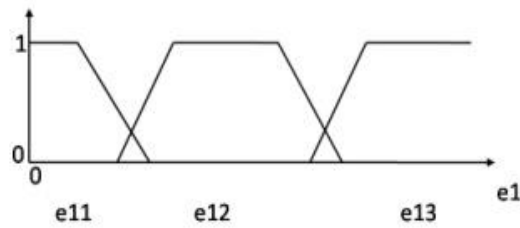


figure 3: Membership function for event (variable) e1

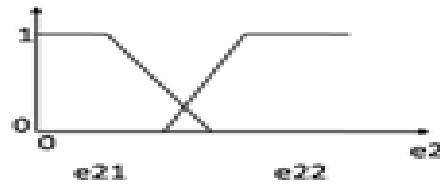


figure 4: Membership function for event (variable) e2

Then for completing this step, for each events (variables) all states of possible linguistic variables are obtained on the Membership function, and for each of the states put a place (the first fuzzification rule). For example, for e1 Membership function we obtain the first step of the Petri net and have shown in figure 5.

Also we can save fuzzy variable value using token in each of the places. And then each of the input linguistic variables using the transition variable to each of the cases may be to connect. This must be done for all events (variables) in the system.

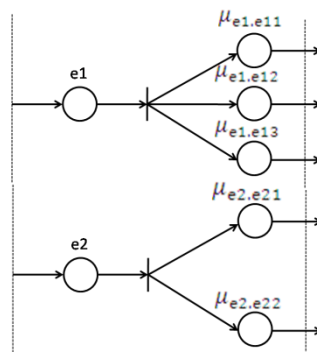


figure 5: calculate Fuzzy value for each variable using the membership function

**Step Two: Calculate the output value of the fuzzy rules**

Create fuzzy Petri net from fuzzy rules and events (variables) that have been calculated in the previous stage. And also reviews condition (calculated rules result) and transfer the results to the next place.

At first, we must review condition accuracy. Therefore, we must create a mapping to do it. For each condition we can create a transition that will review the accuracy of next place results in the net.

Which means that the token with given fuzzy value continues his life cycle. And the token value can be between 0 and 1:

$0 < \text{token value} < 1$

Results at the end of Condition analysis C will be according to the primary values of a fuzzy value.

At first we put the original condition values as token in the places, and then connect equivalent transitions to the place with the edge. Also we will connect transitions that have not been used to the end place.

Then, for each of the rules, the places requiring fuzzy value are connected by the transition to the place that intended for putting rule result, which are shown in figure 6.

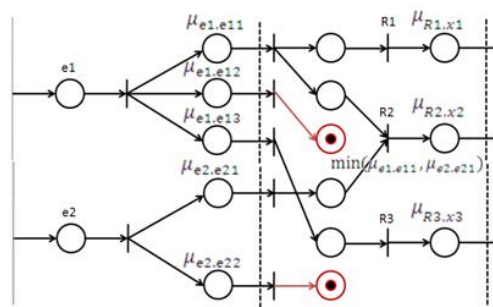


figure 6: second step, calculate value of output rules

### Step three: combine fuzzy rules

Here we have a place for each rule in which a token two-part name of the fuzzy rule and fuzzy value, exists. Here using a transition we put token that have the highest fuzzy value, with the rule number, in the next place which indicate rule that have the maximum fuzzy value. In figure 7 extract a fuzzy rule and that fuzzy value using a transition are shown.

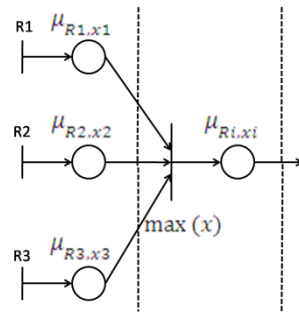


figure 7: extract a fuzzy rule and that fuzzy value between amounts of existing fuzzy rules

#### Step Four: Adding fuzzy state method to fuzzy Petri net

In the previous stages we survey fuzzy state for method inputs (linguistic variables). The values (variables) entered as input to the method were fuzzy. (The first level of fuzzification)

But if your method is a fuzzy state, that means if a fuzzy method is occurring, it will be analyzed at this stage. To do so, we act as follows.

Here we use a transition method to add fuzzy state to the net that support the combination of rules fuzzy state (the first level fuzzification) and method fuzzy state (second-level fuzzification) that calculated as following:

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

Obviously, the fuzzy value of occurrence method, can take value between zero to one.

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

Also, if the method has not a fuzzy state, we can renunciation this stage, or put 1 value instead of the method fuzzy value.

$$\mu_{method} = 1$$

figure 8 shows adding a second level fuzzification of a membership function to the existing net.

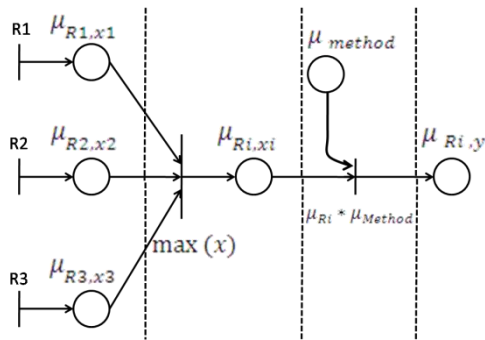


figure 8: Adding fuzzy state to the running method

step Five: continuing or ending of token life in Fuzzy Petri net

Since after adding the Fuzzy method state to the network, the Fuzzy variable (token) may take some value that in the network has not the condition to move. We must move token outside the network. And display of the token life cycle to get out of the show, we use the final location. Petri network to show that a token will be getting out or staying in the life cycle is shown in figure 9.

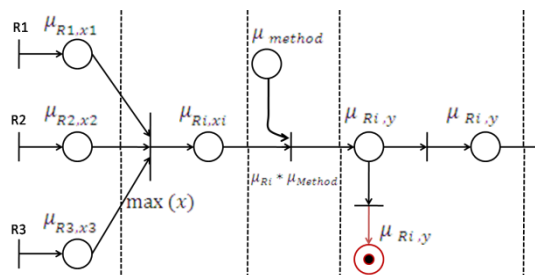


figure 9: Reviews more or end of life token

Step six: Fuzzificating and calculating the final value of fuzzy Petri net

As the fuzzification in primary and middle stages of inference can help to evaluation rules, the final results of the combined output of all rules must be as a crisp value. Therefore, it should be defuzzification, instead of the range of output values, a certain amount of fuzzy set obtained.

There are various methods for defuzzification. Center of gravity method ("COG"), Center of Area method ("COA"), Center of Sums method ("COS"), Mean of Maxima method ("MOM") and sugno defuzzification method. Here we use Center of gravity method.

In the following Center of gravity method is shown.

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



In the previous stages, we just do operations in fuzzy value. But at this stage we must convert created fuzzy values to ultimate tangible values. Means should map the fuzzy values to the membership function and extract the ultimate values (linguistic values) from them.

In this stage use antecedent Fuzzy values for Measure consequent.

Transition used here to use correct antecedent value for consequent membership function.

Doing this is displayed by transition in figure 10.

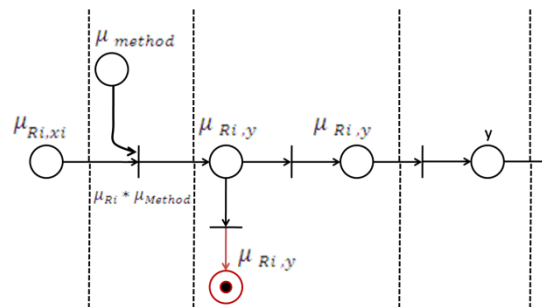


figure 10: fuzzification Process from Fuzzy Petri net

Since the result of combining fuzzy rules together like figure 9, will be fuzzy set of combined outputs and because the final output of fuzzy systems must be a crisp value, you can use the defuzzification operations.

Input of the fuzzification process, is a fuzzy set while the output is a crisp value.

the benefits of using the proposed algorithm

Briefly, the advantages of participation in the proposed algorithm can be found in this paper would name as follows.

- I. Injecting Uncertainty in modeling information system
- II. Using a standard language and the advantages of compatibility with the different sectors towards software development
- III. Ability to implement the method in complex systems
- IV. Formalization of the semi-formal Fuzzy UML model for using the benefits of formal models
- V. Automatic map building in the Fuzzy UML sequence diagram to Fuzzy Petri net

Case Study

Fuzzy Petri net for sequence diagram of a clothes washing machine

We want design a Fuzzy Petri net for clothes washing machine sequence diagram, that by regarding the rate and the amount of being greasy and dirty the time will be set to wash clothes. Then a method to add other participants dress redolence substance is called a fuzzy method that will work (second level of fuzzification). For example, the amount of clothes being fragrant is fuzzily and equal 0.75.

In figure 11 Washing machine system Sequence diagram are shown, in which the input linguistic variables are fuzzy.

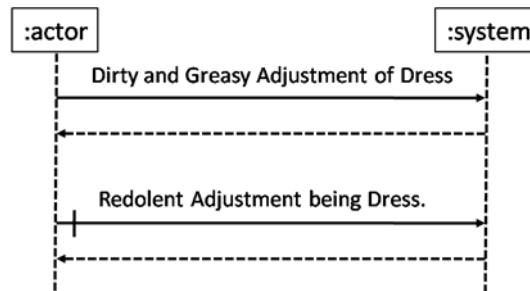


figure 11: System sequence diagram for Washing Machine Case

To do this, first must diagnose all rules - statements - events (variables) Fuzzy / crisp in the system. figure 12 shows rules of the first method.

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

figure 12: Rules of the first method

Also we must design fuzzy membership functions for each of the events (variables). Obviously, the membership functions design must be done by an expert. figure 13 shows membership functions for the events (variables) antecedent or consequent fuzzy.

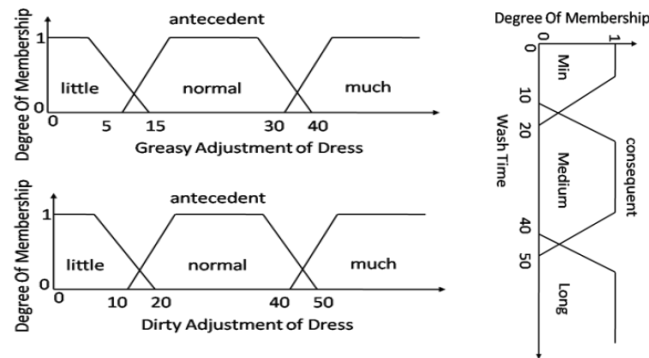


figure 13: Display membership functions for fuzzy variables, the first method

In the first method the inputs were linguistic variables and fuzzy, but the method was not fuzzy state. Then another method (second method) is called in which linguistic variable "Redolent" will be added to the system. And the amount of the aromatic Article is diagnosed. figure 14 shows the rules of the second method. In this method (second method) the method of input variables and the method state is fuzzy.

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

figure 14: Rules of the second method

Also we must design membership functions for each events (variables) of Fuzzy second method. figure 15 shows membership functions for fuzzy events (variables) of antecedent or consequent second method.

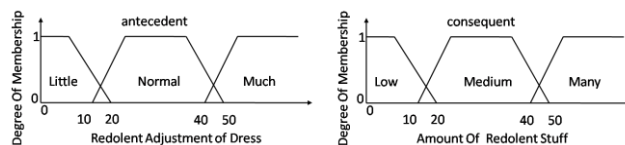


figure 15: Display membership function for fuzzy variables, the second method

Now having all the required rules and also required membership functions we can creat fuzzy Petri net for sequence diagram clothes washing machine, using the steps described earlier established which are shown in figure 16.

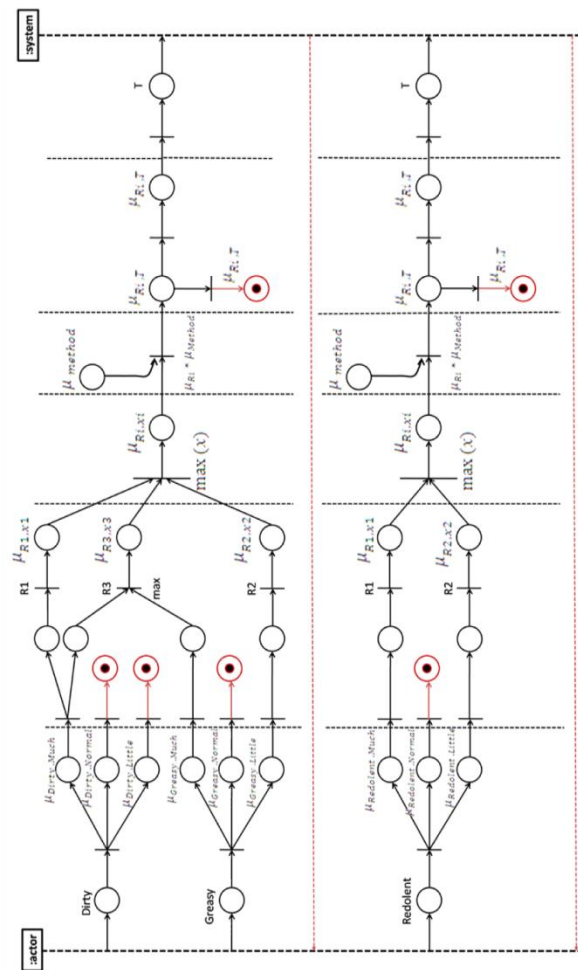


figure 16: Fuzzy Petri net for clothes washing machine sequence diagram

### 3. Conclusions

Changing the semi-official Fuzzy UML models to formal models feature Fuzzy Petri net symptoms such as formulating graphical processing and analysis tasks automatic offers. In this paper has been tried to model fuzzy formal Petri net in the Fuzzy UML sequence diagram, to access the capabilities of the formal models. It can provide capabilities such as automated processing and formulating of graphical symbols and analysis task. Because before the implementation fuzzy the system, calculated non-operational factors such as system reliability and error tolerance and etcetera was not possible and a prototype system was built to the above factors should be calculated.

Here it is tried to automatically mapping the Fuzzy UML sequence diagram to Fuzzy Petri net to step to calculate these parameters before the initial prototyping and to occur automatically. This automatically building reduces the cost and evaluation prototypes systems. That these efforts will be reduce the software construction costs.

---

**References**

- [1] Faul B.M., "Verifiable Modeling Techniques Using a Colored Petri Net Graphical Language", *Technology Review Journal*, 2004.
- [2] Shin M., Levis A., and Wagenhals L., "Transformation of UML-Based System Model into CPN Model for Validating System Behavior", In: *Proc. of Compositional Verification of UML Models, Workshop of the UML'03 Conference*, California, 2003.
- [3] Bernardi S., Donatelli S., and Merseguer J., "From UML Sequence Diagrams and Statecharts to Analysable Petri Net Models", *ACM Proc. Int'l Workshop Software and Performance*, 2002.
- [4] Eshuis R., "Semantics and Verification of UML Activity Diagrams for Workflow Modelling", Ph.D. Thesis, University of Twente 2002.
- [5] Pettit R. G. and Gomaa H., "Validation of dynamic behavior in UML using colored Petri nets' UML", Zaragoza, Spain, p.295-302, 2002.
- [6] Saldhana J., and Shatz S. M., "UML Diagrams to Object Petri Net Models: An Approach for Modeling and Analysis", *Proc. of the Int. Conf. on Software Eng. And Knowledge Eng. (SEKE)*, Chicago, p.10 – 103, 2000.
- [7] Elkoutbi M., and Rodulf Keller K., "Modeling Interactive Systems with Hierarchical Colored Petri Nets", *1998 Advanced Simulation Technologies Con.*, Boston, p.432- 437, 1998.
- [8] Bernardinello L., De Cindio F., "A Survey of Basic Net Models and Modular Net Classes", *LNCS*, vol 609, Springer-Verlag, 1992.
- [9] S. Balsamo, et al "Model-Based Performance Prediction in Software Development: A Survey", *IEEE Transactions on Software Engineering*, 30(5), MAY 2004.
- [10] Merseguer J., LopezGrao J. P., Campos J., "From UML Activity Diagrams To Stochastic Petri Nets:Application To Software Performance Engineering", *ACM, WOSP 04 January*, 2004,
- [11] Fukuzawa K., et al, "Evaluating Software Architecture by Colored Petri Net", Dept. of Computer Science, Tokyo Institute of Technology Ookayama 2-12-1, Meguro-uk, Tokyo, 2002
- [12] Merseguer J., Bernardi S., Campos J., and Donatelli S., "A Compositional Semantics for UML State Machines Aimed at Performance Evaluation", *Proc. of*

the 6th Int. Workshop on Discrete Event Systems (WODES'02), Zaragoza, Spain, p.295-302, 2002.

[13] Motameni H., et al, "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, december 4\_14 2006.

[14] Motameni H., et al, "Using Markov Theory For Deriving Non-Functional Parameters On Transformed Petri Net From Activity Diagram", proc of software engineering conference (russia), November 2006.

[15] Motameni H.,Zandakbari M., and Movaghar , "DERIVING PERFORMANCE PARAMETERS FROM THE ACTIVITY DIAGRAM USING GSPN AND MARKOV CHAIN ", ICCSA 2006, Proceeding of 4<sup>th</sup> International Conference On Computer Science and Its Aapplications, 2006.

[16] Motameni H., "Evaluating UML State Diagrams Using Colored Petri Net", SYNASC'05, 2005.

[17] Motameni, H et al. "Verifying and Evaluating UML Activity Diagram by Converting to CPN ", Proc of SYNASC'05,Romania,Sep 2005.

[18] Object Management Group, "UMLTM Profile for Schedulability, Performance, and Time Specification", OMG Document, Version 1.1, January 2005.

[19] Rumbaugh j., Blaha m., Premerlani W., Eddy F., Lorensen W., "Object-Oreinted Modeling And Design", prentice hall , Englewood Cliffs,USA.

[20] Wang lu, "Fuzzy UML", Seminararbeit, Sommersemester, 2005.

[21] Zongmin Ma., "Fuzzy Information Modeling With the Uml", Idea, 2005.

[22] Ma Z.M., "Extending UML For Fuzzy Information Modeling In Object\_Oriented Database ", theories and practices, idea group publishing, 2004.

[23] terry quatrain, "visual modeling with rational rose 2000 and UML", Addison Wesley, 1999.

[24] Burcin Bostan-Korpeoglu, Adnan. Yazici, "A fuzzy Petri net model for intelligent databases", Department of Computer Engineering, Middle East Technical University, Elsevier Ankara, 2006.