Transforming FUZZY Collaborated Diagram to Fuzzy Petri Net

H. Motameni
Sama Technical and Vocational Training School, Islamic Azad University, Sari Branch, Sari, Iran
Motameni@iaeusari.ac.ir

M. Fallah
Islamic Azad University, Science & Research Branch, Tehran, Iran
malihe.fallah@yahoo.com

H. Nematzadeh
Faculty of Computer Sciences & Information Systems Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
nhossein3@live.utm.my

Abstract- UML¹ is a semi formal language and many researches have been performed to transform this language into formal methods including Petri nets. Thus, verification and validation of the qualitative parameters could be achieved. Since the majority of the real world information is uncertain, the fuzzy UML diagram has been extensively used by system analyzer. This paper attempts to transform system collaboration diagram created in fuzzy UML into fuzzy Petri net.

Keywords: Collaboration diagram , Fuzzy uml , Fuzzy petri net

I. INTRODUCTION

Nowadays, UML diagrams are extensively used in software design. 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. It is obvious that the lack of this important ability in UML models remains the needs of the costomer and the market. This fact makes this researches more important. Several works have been performed to tackle with the semi-formal problem of UML. Some of these have only used a transformation algorithm, which transforms the created UML model into a Petri net as a mathematical and formal model, which in turn, contains the visual aspect of modeling and pursues the verification operations with further ability [1,3,5]. However little work has been done on the Fuzzy UML , as it is a new model. In [2] activity diagram in Fuzzy UML has been transformed to Fuzzy Petri net. In another work [1] a case tool to obtain qualitative parameter from GSPN using continious time Markove Chain (CTMC), has been designed. In [4] UML model has been transformed to colored Petri net, which allow to analyse using simulation techniques. In [5] the transformation of activity diagram in UML to Generalized Stochastic Petri Net has been showed. In [2] the transformation of UML diagram to Petri net model, which allow the performance evaluation of the system, has been showed. Works has been done to transform software models to colored Petri net, which showed a better model than UML [4].

In this work we have attempt to present a pattern to transform fuzzy collaboration diagram to fuzzy Petri nets, which allow us to add the formal model performance to Fuzzy UML. With this mapping we were able to find a way for verification and evaluation operation in the first cycle of the software production.

II. FUZZY COLLABORATION DIAGRAM

The collaboration diagram is used for modeling of the system dynamic behavior. This diagram shows the reaction among the objects. A simple collaboration system includes: actor, system, and messages. Each message itself includes events and conditions. This diagram uses fuzzy rules for transforming the state of an object to another. A fuzzy rule can be expressed as in eq. (1)

\[
\text{Rule} = \text{If<condition list>Then<event list>} \quad (1)
\]

III. FUZZY PETRI NET

We introduce the following fuzzy Petri net(FPN) structure to model fuzzy ruler [6-8]:

\[
(P,P_s,P_e,T,TRF,A,I,O,TT,TF,PR,PPM,TV)
\]

1. \(P\) is a finite set of fuzzy places. Each place has a property associated with it, in which
   - \(p_s\subseteq P\) is a finite set of input places for primitive events.
   - \(p_e\subseteq P\) is a finite set of output places for actions or conclusions.

2. \(T\) is a finite set of fuzzy transitions. They use the values provided by input places and produce values for output places.

3. \(TF\) is a finite set of transition functions, which perform activities of fuzzy inference.

4. \(TRF:T\rightarrow TF\) is transition type function, mapping each transition \(\in T\) to a transition function \(\in TF\).

5. \(A\subseteq (P\times T\cup T\times P)\) is a finite set of arcs for connections between places and transitions. Connections Between the
input places and transitions \((P \times T)\) and connections between the transitions and output places \((T \times P)\) are provided by arcs. In that:

- \(I: P \rightarrow T\) is an input mapping.
- \(O: T \rightarrow P\) is an output mapping.

6. \(TT\) is a finite set of fuzzy token (color) types. Each token has a linguistic value (i.e., low, medium and high), which is defined with a membership function.

7. \(O: T \rightarrow PLs\) token type function, mapping each fuzzy place \(\in P\) to a fuzzy token type \(\in TT\). A token in a place is characterized by the property of the place and a level to which it possesses that property.

8. \(AEF:Arc\rightarrow Expression\) is arc expression function mapping each arc to an expression, which carries the information (token values).

9. \(PR\) is a finite set of propositions, corresponding to either events or conditions or actions/conclusions.

10. \(PPM: P \rightarrow PR\), is a fuzzy place to proposition mapping, where \(|PR| = |P|\).

11. \(TV: P \rightarrow [0,1]\) is truth values of tokens \((\mu)\) assigned to places. It holds the degree of membership of a token to a particular place.

IV. TRANSFORMATION ALGORITHM

First step: Calculation of Fuzzy value for each variables using membership function. For each message rules and system condition after operation and its membership function for all variables (fuzzy and certain) could be evaluated.

Second step: Calculation of the rules resultant in fuzzy forms. First we need to check the correctness of the conditions, therefore we should provide a mapping to do that. For each condition we provide a transition that is responsible to check the correctness of it and the result will go to another place. That means the token with the given fuzzy amount continues its lifecycle. And the token may have the value from 0 to 1, that means:

\[
0 \leq \text{token value} \leq 1.
\]

First the primarily amounts of the rules is consider as a place and tranion of their equivalence through arc joint to places. Also unused transiton are joined to a final places.

Third step: Sometimes we have more than one condition, so depends on fuzzy logical operations we choose the appropriate function. The concept is in equations in (2):

\[
\begin{align*}
\text{OR} & = \mu A(x) U \mu B(x) = \max(\mu A(x), \mu B(x)) \\
\text{AND} & = \mu A(x) n \mu B(x) = \min (\mu A(x), \mu B(x))
\end{align*}
\]

Here, using a transion, we can send the token which has the highest amount of fuzzy, with the number of the corresponding rules to another place.

Fourth step: Addition of state of the fuzzy method to the fuzzy Petri net. If the method itself is a fuzzy method, we do as follow: Using a transition, the fuzzy state of the method is added to the net, which its resultant is a fuzzy state comprising a sum of fuzzy state of the rules and fuzzy state of the method.

This is done by the following equation (3):

\[
\mu_{\text{method}} = (\mu_{\text{method}} + \mu_{\text{rules}})
\]

It is obvious that the amount of event of the fuzzy method could be an amount between 0 and 1.

Fifth Step: Evaluation continuation or the end of life of the Token in the Petri Fuzzy condition. In this step we first evaluate that the token has the ability of the function or it should be sent to a final place. For this we use from a logical OR.

Sixth Step: Defuzzification and the evaluation of the final amount of the Fuzzy Petri net. The final result after the combining the output of all rules should be in the form of certain amount. So it should be defuzzified. The defuzzification process is done based on center of gravity test. Center of gravity can be expressed as equation (4).

\[
\text{COG} = \frac{\int_{a}^{b} \mu A(x) x \, dx}{\int_{a}^{b} \mu A(x) \, dx}
\]

In this step we should transform the resulting Fuzzy amount to the final crisp number. It means that the Fuzzy amount should be mapped on the membership function, and extract the final value from it.

V. CASE STUDY

Making the Fuzzy Petri Net for the diagram of the collaboration of the evaluation of the concrete composition. We want to design a Fuzzy Petri net for a collaboration diagram of a concrete drop mixer in Laboratory scale. Which could evaluate the usefulness of the concrete mixture in the case of the reduction of concrete volume. This also state the amount of sand and grit change in the mixture.

For this we should first recognize all the rules, conditions, and the events existence in the system.
We should also design all the Membership function, for each Fuzzy event. It is obvious that the design of the membership function should be done by an expertise.

Now with all the needed rules and neede membership function could design the Fuzzy Petri Net for collaboration diagram of the concrete equipment. This is depicted in the fig 6.

We now want to show the evaluation extracted result, by entering the desired variables to the fuzzy Petri net resulted from collaboration diagram.

For example, lets assume that the concrete drop be around 6 inches, and assuming the correctness of the equipment function, we expect from this net, that the amount of the grits in concrete be evaluated from all the input variables used.

\[
\begin{align*}
\mu_{\text{Shake}}(6) &= 0 \\
\mu_{\text{Normal}}(6) &= 1 \\
\mu_{\text{Little}}(6) &= 0
\end{align*}
\]

From fig 7, it is obvious that we could calculate the evaluate the certain amount from fuzzy values from the input of variable.

---

**VI. CONCLUSION AND FUTURE WORK**

In this work attempts have been made in the transformation of the collaboration diagram in Fuzzy UML to Fuzzy Petri Net. In future work we could work in the evaluation of qualitative parameter of the Fuzzy Petri Net. We could also prepare a mapping of existent diagram of Fuzzy UML to Fuzzy Petri Net. For this we should first transform the Fuzzy UML diagram to formal model, using fuzzy Petri net.

**REFERENCES**


