Modeling and Evaluation of Information Systems Using Fuzzy Logic

Marjan Abdeyazdan¹, and Sayyed Hedayat Tarighinejad²

Abstract—Unified Modeling Language (UML) can be used to address functionality needs in software engineering. Although Object Management Group (OMG) proposed a performance index in 2002 to support non-functionality needs, it is necessary to have a formal model to evaluate performance. Petri Net by using mathematical support is useful in behavior modeling. On the other hand, due to the irrational nature of data in information systems, using linguistic variables and fuzzy logic will have higher accuracy in calculations of assessing system performance and efficiency. In this paper, by converting the real model into a formal model (F_CPN), we make it possible to evaluate the system performance with a formal model. By using fuzzy logic and feedback analysis before implementation we can improve the performance of system. By analyzing the results we will see that fuzzy approach makes performance improve.


I. INTRODUCTION

TODAY the widespread use of information systems in different applications is undeniable. By the growth of software we can divide the proposed methods in system development into two categories including structured and object-oriented. Object-orientation has made a revolution in software development and it considered the expressed ideas in structured approaches with a new attitude. UML, as a modeling language in support of object-orientation, was proposed in 1990s. Unified symbols, techniques and instructions in UML have made it a public language for different applications in object-orientation. Today although UML has become widespread, two defects and weaknesses are observed in it:

- The techniques used in UML are able to address definite matters. On the other hand, uncertainty can be seen in many information systems. Since information systems are proposed to meet the needs of users and uncertainty is sometimes noted due to the natural nature of users’ needs, so if one can insert uncertainty in UML, better utilization of this language will be provided.

- Since UML is not a formal model, evaluation of software systems is not directly possible. Although OMG has undertaken the responsibility of UML development, it introduced and index for supporting performance concepts[ 3]. Therefore, in order to evaluate performance, it is essential to convert the real model into a formal model.

In the second part of this paper, the context of this work will be considered. This section focuses on Fuzzy Petri Net. In the third section of this paper, works done in this area will be reviewed. The proposed solution for presenting a formal model and its analysis will be considered in the fourth section. In section five a case study will be presented and the sixth section will end in conclusion.

II. CONTEXT

This field contains Unified Modeling Language, Fuzzy Logic, Petri Nets and Fuzzy Petri Nets. Given the importance of this subject, the latter two issues will briefly be reviewed.

A. PETRI NETS

These nets are used as a tool for modeling systems (especially discrete event systems[4]) . Among formal models, Petri Nets are such powerful tools for simultaneous modeling. By using Petri Nets, which are based on graph theory, one can analyze systems with asynchronous, indefinite and random structures.

A.1. COLORED PETRI NET

Kurth Jensen proposed this net in his PhD thesis in 1980. Different colors in Colored Petri Net make it possible to transit the tokens or nuts that carry data values and are distinguishable. In these nets, in order to fire a transmission the existing tokens in input places should coincide with the terms on arcs. Colored Petri Nets have high ability in presenting the concept of concurrency by having colored tokens.

A.2. FUZZY PETRI NET

The difference of the above mentioned Petri Nets lies on those components of Petri Net that have become fuzzy. Place, token, transition (and arc, with less importance) are the components of Petri Net that make it possible to perform fuzzy logic on them. Fuzzy token represents the degree of belongingness of the token to the place. This value is normal in Petri Net {0, 1}, but in Fuzzy Petri Net each value can be between[ 0, 1]. A fuzzy place expresses each statement (term) as fuzzy. A fuzzy transition expresses the accuracy of a fuzzy produced rule as if-then. Each
transition has a threshold limit that determines the least amount that a token can fire with.

III. RELATED WORKS

Today a great deal of research has been done in transforming the real model into a formal model. Among formal models, Petri Nets[5] and Queue Net[6] and among UML diagrams, phase diagram[7] have been the focus of attention among the activities done in this field. Among these activities, Merseguer[5] and Shatz[8] approaches have high capabilities. These two approaches have focused respectively on Labeled Generalized Random Petri Net and Object Petri Net.

Merseguer in[5] revised the Standard Function Profile presented by Object Management Association and considered UML diagram maps in software function.

Shatz and Saldhana in[8] transformed phase diagram into an Object Petri Net and then presented a Colored Petri Net of the whole system based on it.

However, none of the above papers have considered the use of function profile in analyzing Fuzzy Petri Net for evaluating the system performance.

IV. PROPOSED METHOD

In the first section, two weaknesses of UML in addressing issues were mentioned. Figure 1, shows an algorithm process for addressing the mentioned weaknesses. Addition of fuzzy logic to UML is focused in[2]. Class, use case, order and activity diagrams are designed as fuzzy and a sub-profile of F-UML is considered for modeling software performance. What is emphasized in this paper is presenting a formal model of F-UML diagram and analyzing them before implementation (gray parts in Figure 1).

A. USING FUZZY COLORED PETRI NET IN PRESENTING PERFORMANCE MODEL

In 2002, OMG presented several profiles for expanding UML. SPT profile was introduced for its capabilities in Timing, Performance (Efficiency) and Time characteristics. Efficiency profile, like other profiles, is used for supporting extension, stereotypes and labeled values. Each profile includes different stereotypes. Labeled values are some features of stereotypes in the profile and are linked to the model’s stereotype element as descriptions.
Several actions can use the same resource in the system, so service rate (fire) is different for actions. In the queue system, it is shown by chains but in Colored Petri Net it is shown by different colors relating to the tokens.

B. THE PROPOSED ALGORITHM

F-CPN model is composed of T transitions and C colors that show the number customers’ classes. Transitions in F-CPN show the resources of UML model. For each factor with the stereotype <<PAclosedLoad>>, an additional transfer is defined in F-CPN that represents the time between the end of an interaction with system and beginning of the subsequent repetition (e.g. user think time). Different factors have also various workloads in the system. In such a case, F-CPN model is divided into several sub-models of F-CPN that are independent of each other. Each sub-model has its own workload. The requests belonging to a sub-model have multiple classes. The available classes in each sub-model are displayed with different colors in F-CPN model. The color of each sub-model is unique to that sub-model and is not repeatable in other sub-models.

An Open Petri Net is a net with input/output to external environment. Closed Petri Net doesn’t have any contact with external environment. Open and Closed Petri Nets respectively support <<PAopenLoad>> and <<PAclosedLoad>> stereotypes in Use Case Diagram. By using Activity Diagram we can obtain Routing Matrix of Fuzzy Petri Net (transition probability matrix). It should be mentioned that multiple actions in a system can use the same resource. To support this, we define the following features in F-CPN model:

Supposing that system resources are presented as RES= \{res_1, res_2, ..., res_m\}. Obviously, we will have Counter(res)_i=m. For each res_i \epsilon RES, we label all the actions in the set \{act \in ACT | resource(act) = res_i\} with a unique number in interval [1, 2, ..., Counter(res)]. This unique number is indicated by indicator(act) feature (for each action act \in ACT we define indicator(act)). Input in Open Petri Net enters the transition which the initial action of Fuzzy Activity Diagram uses. Input rate in Fuzzy Petri Net equals with PA occurrence tag in a factor that uses Fuzzy Activity Diagram for its use case. Transition probability in Fuzzy Petri Net results in the corresponding label PA probe in transiting Fuzzy Activity Diagram.

The algorithm of transforming F-UML model into F-CPN model for evaluating the software efficiency is as the following (suppose that factor X is annotated with stereotype <<PAopenLoad>>).

1) Calculate the following features:
   - Counter(res) \forall res \epsilon RES,
   - Indicator(act) \forall act \epsilon ACT,
   - C=maximum res \epsilon RES | Counter(res)| and T
2) Calculate Transition Probability Matrix.
3) SR[i, r] (service rate for customers with class r in transition i) is calculated as the following (m is an action in Fuzzy Activity Diagram):
   \[ SR[i, r]=rate[r]/demand[act] \]
   where \( i=\text{identity}[\text{resource}[m]] \), \( r=\text{indicator}[act] \)
4) Vector \( \lambda[r] \), that shows arrival rate for customers with class r, is calculated as the following:
   \[ \lambda[r]=\text{arrival rate}[X] \]

In other words, arrival rate is calculated based on use case (PAoccurrence tag) that results in using Activity Diagram.

If X is annotated with stereotype <<PAclosedLoad>>, the following changes will be applied in the above algorithm:

- The number of transitions is one plus available resources in Deployment Diagram.
  \[ T=|\text{RES}|+1 \]
- The added transition is indicated by zero and as we mentioned before, it is considered to address delay (e.g. thinking time). Fire rate of this transition is as the following for all classes of customers (s is a customers’ classes):
  \[ \text{SR}[0, s]=1/\text{extdelay}[X] \]
- N is the number of available requests in the system and is calculated based on PA population tag. The above label (tag) in Petri Net is displayed by population feature. We will have:
  \[ N=\text{population}[X] \]

By combining these two algorithms, one complete algorithm is obtained that the number of F-CPN sub-models
equals A+B in it. A and B are respectively the number of factors with stereotypes <<PAopenLoad>> and <<PAclosedLoad>>.

In order to support fuzzy approach in the algorithm, corresponding to belongingness degree of fuzzy transition in Fuzzy Activity Diagram, we add one parameter to fuzzy transition of F-CPN model. Fuzzy token explains how to use resources as fuzzy (resource is not completely in control of action).

V. CASE STUDY

This case study is dedicated to the facility sector in an imaginary bank. Figure 2 shows two use case diagrams with only one use case. Deployment diagram of the system is displayed in figure 3. Figure 4 shows an action of activity diagram those results in use case. The above activity diagram has 11 actions.

The middle graph, showing resources and the number of them, can be observed in figure 5. In this graph, for each action of activity diagram there is one node. By numbering the available resources in deployment diagram, it is seen that when one resource is reused, one additional unit adds to the indicator. Transition probability is seen on the edges of the mentioned graph.

Figure 6 shows Colored Petri Net of the middle graph. Supposing that the use case of activity diagram is labeled as PA occurrence="exponential",10], then we will have: λ=1/10

“PA demand” for those actions that use Application Server resource has respectively the following values (Colored Petri Net has 5 classes):

\{0.2, 0.1, 0.5, 0.3, 0.2\}

Fire rate in transition of Petri Net for Application Server resource is:

\[SR = \text{rate}_{\text{res}} / \text{demand}_{\text{act}} = (25, 50, 10, 16.666, 25)\]

By obtaining arrival rate (λ) and fire rate (service), we can analyze Petri Net.

Implementation tool for charting UML diagram is Rational Rose (ver7) and for network stimulation we use CPN. Calculations of fuzzy logic are done by MATLAB. After stimulation, CPN provides meters analysis related to efficiency. In this case study, three meters are focused including productivity, operational strength and queue length. Obviously, the number of tokens in each place shows queue length.
VI. CONCLUSION

In this paper, we proposed a novel strategy for evaluating software systems. Presenting a formal model (Fuzzy Colored Petri Net) based on the real model (F-UML) makes it possible to consider non-functional needs of software for developing systems along with software process models. Simulation results will lead to evaluation of this model before implementation of the system. Using fuzzy approach will also increase efficiency and performance of system (improvement of queue length).

VII. ACKNOWLEDGEMENT

This research supply by Mahshahr branch, Islamic Azad University, Iran. I would like to thank to the board and jury of Mahshahr University and appreciate the most.
REFERENCES

http://dx.doi.org/10.4018/978-1-59140-384-5
http://dx.doi.org/10.1007/978-3-540-24663-3_13
http://dx.doi.org/10.1007/978-0-387-35520-7_6