MEDWELL PUBLISHING
International Journal of Soft Computing 4 (1): 32-38, 2009
ISSN: 1816-9503

Behavior Modeling in Uncertain Information Systems by Fuzzy-UML

[1]Ali Haroonabadi and [2]Mohammad Teshnehlab
[1]Azad University, Science and Research Branch, Tehran, Iran
[2]Department of Electrical Engineering, Khaje Nasir Toosi University, Tehran, Iran

Abstract: Because of the natural essence of requirements, uncertainty in information systems is unavoidable. To develop the systems (analyze, design, implementation, test) object oriented methods describes their concepts, with UML language. UML as a standard object oriented language has ability to investigate the different aspects of behavioral and structural in software engineering. If enter the uncertainty in UML, the extended version named Fuzzy-UML will be produced. This developed version includes both of structure and behavior sights. Because of the lack of UML ability support for evaluation, the necessity of transforming the pragmatic model of (UML) system to the formal model (Petri Nets) is considered (because of being uncertain of the information, the pragmatic model transforms to the Fuzzy Petri Net formal model). We will have the ability of evaluating the complex systems, with preparing the proper model of system.

Key words: UML, fuzzy system, software performance evaluation, software development model, software engineering

INTRODUCTION

Now a days the extent of Information systems in the different usages is undeniable. By growing the software, the propounded methods for development the systems are divided into 2 groups: structural and object oriented. Object orientation created a great transformation in software development and it considered the propounded ideas in structural approaches with a new look: Methodologies, programming languages, Data bases (OQL instead of SQL) and even Markup language (for example: XML that we can name it an object oriented version of HTML) can be considered as examples for object oriented thought in usage. In the other hand for describing the concepts, the selected language performs a basic role and because of the severalty of looks in the structural approach, this subject faced to lots of challenges. UML as a Unified Modeling Language (that it was supporting the object oriented concepts), proposed by Rumbaugh, Booch and Jacobson in the middle of 1990 decade. The unification of symbols, techniques and propounded instructions in UML, introduced it as a general language for different usages in object oriented. Although nowadays UML has been extended a lot, but two lacks are considered:

- The used techniques in UML are useful for certain problems. In the other hand uncertainty is considered in the lots of information systems. Because Information systems resolve the users requirements and uncertainty is considered cause of user requirements natural essence, so if we enter the uncertainty in UML, the causes of better exploitation will be prepared.
- Because UML is not a formal model, software systems evaluation is not possible by it directly. Although, OMG (2002) that is responsible for UML development, introduced a profile to support performance concepts, but it needs to transform pragmatic model to formal model, for performance evaluation.

The proposed idea in this study (for conquer the above problems) is considered in Fig. 1.

In the selected approach of study, UML diagrams in addition to supporting the uncertain functional requirement are used to specify the software performance requirements. The Software Performance Engineering (SPE) community introduced UML as a reference of describing the software systems in its international workshop (Smith and William, 2000). In Smith (1990), SPE is definition like a method (a quantitative and systematic approach) to produce the software systems in away to observe the quality objectives. We review the SPE approach that lots of activities has been done on it, in this study. In additional to introducing F-SPE (Fuzzy-SPE) method, what is important in this study, is specifying the role of some of F-UML diagrams (use case, sequence) in performance from the object oriented aspect.

Corresponding Author: Ali Haroonabadi, Azad University, Science and Research Branch, Tehran, Iran

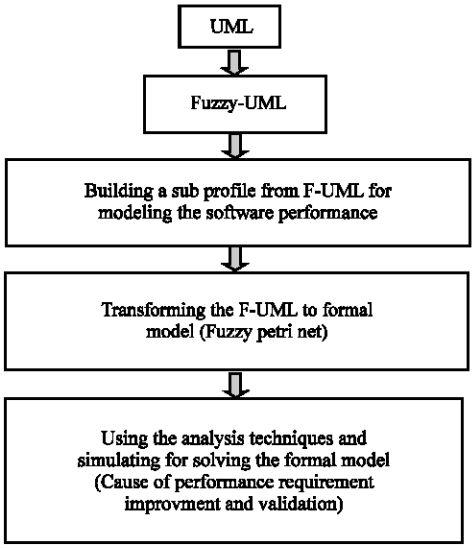

Fig. 1: Presented idea in this study

In the second part of study, we talk about entering the uncertainty in UML and review the fuzzy subjects from 2 aspects of information systems: structural and behavioral. You can refer to Ali and Teshnehlab (2007), Ali (2006) and Zongmin (2005) to consider the additional subjects for this section. The third section introduces a sub profile from F-UML to modeling the software performance, that it helps to transform the F-UML to the uncertain formal model (fuzzy Petri Net) and we can improve the performance requirements and system functionality after solving the formal model.

PRESENTING A FUZZY MODEL TO DEVELOPING THE UNCERTAIN SYSTEMS

UML supports both behavioral and structural aspects of system, so fuzzy concepts are propounded in the present study: fuzzy structure and fuzzy behavior. To supporting system fuzzy structure, a fuzzy data model and to supporting system fuzzy behavior, the models are needed, which can support system functionality fuzzily.

The first part of this study is about the data modeling, that we enter the uncertainty in data structure by presenting a class diagram fuzzily. The fuzzy behavior modeling is inspected in the second part. This subject is realized by presenting the use case and sequence diagrams fuzzily. The other diagrams have been inspected in Ali (2006) and Ali and Teshnehlab (2007).

F-UML data model: The class diagrams in UML are the logical models and they describe the system main structure. The classes and the relationships among classes are the composed elements of the class diagram. By entering the uncertainty into these elements, the F-UML data model is produced. Considering that Zongmin founds (Zongmin, 2005), In the context of classes, the three levels of fuzziness are defined as follows:

- Fuzziness in a way that the class could be propounded fuzzily in the data model, by notice to the contains of it (in attributes section).
- Fuzziness in a way that do the objects belong to the class? Even with supposing that the class structure is certain, some of the objects can belong to the class with membership degree between 0 and 1.
- The third level of fuzziness is propounded on the class instance attribute values. An attribute in a class is defined on a domain, which is supposed for its values. When this domain is a fuzzy subset, the fuzziness will appear on the attribute value.

The attribute or the class name in the first level should be described by this phrase "WITH mem DEGREE"
where, $0 \leq men \leq 1$. This value shows the degree of belonging the attribute to the class or the class to the data model.

In the second level of fuzziness, the membership degree of an instance from the class that it belongs to the class should be specified. So an addition attribute in the class is defined for representation the instance membership degree to the class, which its domain is [0,1]. This special attribute has specified with. The classes with μ the second level of fuzziness have specified by a rectangle that its lines are dash.

In the third level, a fuzzy keyword is appeared in front of the attribute.

Figure 2 shows the banking account fuzzy class. In the mentioned class the credit account attribute can have the fuzzy value (the third level of fuzziness). In the other hand the credit account attribute is a linguistic variable and it has a domain like fuzzy sets (for example: little/much).

In addition the account type, also specifies the "credit account" attribute membership degree to the class (the first level of fuzziness): "Credit WITH 0.8 membership DEGREE".

Also, we propound the μ attribute to showing that the object is the class instance (the second level of fuzziness).

The relations among the classes are divided into 4 groups and they are propounded fuzzily (Ali, 2006; Zongmin, 2005). Fuzzy generalization, fuzzy association, fuzzy aggregation and fuzzy dependency.

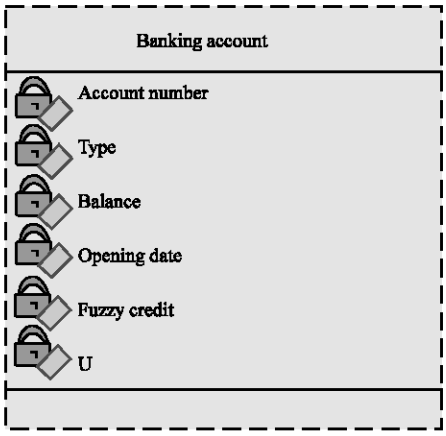

Fig. 2: A fuzzy class of banking account

Presenting a fuzzy model from the system behavior: There are different approaches to describe the system behavior. The system behavior can be presented in the information systems by the used techniques in UML. In addition briefly we extend the steps that UML presents for certain systems development to the uncertain systems and we improve the F-UML version.

Specifying the uncertain requirements of system: For assigning the system behavior, we should specify its functionality requirements at first. The requirements extraction in object orientation is service oriented. When a service is presented uncertainly, the fuzzy use case will be propounded. The showed symbol in Fig. 3 is used to describe this concept in F-UML.

After specifying the certain and uncertain use cases, we perform to drawing the fuzzy use case diagram (Ali and Teshnehlab, 2007).

Assigning the flow of events for realizing the fuzzy use cases: A use case is a sequence of actions that a system does to prepare an observable result for user (IBM, 2003). In the other hand, for each use case we can set the scenarios, which explain the service presenting process of the use case. In UML we use sequence diagram to realizing the use cases. If the selected use case is uncertain, the sequence diagram will be uncertain too. We use the dash lines for uncertain messages representation in mentioned diagram. Figure 4 represents this subject.

The messages in sequence diagram explain the methods. The uncertainty in method has 2 level of fuzziness.

The first one is that the method belongs to the object with membership degree and the second one is that the method essence is fuzzily. In Fig. 4, the method C, which is produced from message C may belong to the object B

Fig. 3: The fuzzy use case symbol in F-UML

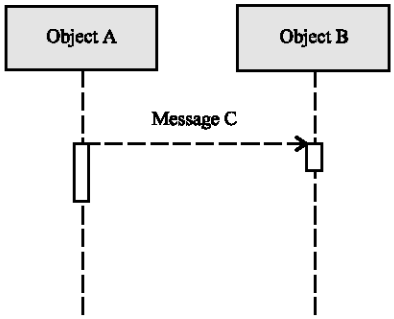

Fig. 4: The fuzzy message symbol in F-UML

with the membership degree between 0 and 1 or it can be an uncertain method itself.

In addition, we will need to make fuzzy decisions in many software systems. The propounded fuzzy scenarios in F-UML present a solution for this subject. With specifying the effective parameters in a scenario, we can transform the scenarios to the compositional fuzzy rules. An example about this subject was mentioned in the fourth section of the study.

F-SPE METHOD

The complete software systems are introduced as the most complex products that were produced by human civilizations (Brooks, 1987). These systems need the formal models to modeling and analysis.

Although, the several process models are proposed in software engineering field, but none of them has propounded the modeling and performance evaluation in software systems at software development life-cycle. It's necessary to say that separate inspection of these 2 subjects is false (Merseguer and Campos, 2004) without considering to the relationship between them. The propounded profile by OMG was a start point to remove the gap between software evaluation techniques and software development processes. The proposed approach in this study, is a step for connect these 2 subjects and it can be used with all of the software development process models.

We notice the role of UML diagrams in system performance computations to receive the F-SPE method. UML diagrams are divided into 3 groups: Behavioral diagrams, structural diagrams and implementation

Table 1: A summary of annotation

	Annotation	Kind*	Referenced value
Use case diagram	Probability that an actor executes a uses case	A	Association link
Sequences diagram	Probability of success of a message	B	Message
	Message size	C	Message

diagrams. All of the above diagrams relate to the system performance aspects. For describing a system performance completely, behavioral and structural diagrams are considered in the modeling level and the implementation diagrams are propounded when the system hardware configuration is considered (Merseguer and Campos, 2004) Considering that the systems performance studies often focus on the system dynamic aspect, so among the above diagrams, behavioral diagrams have the most role in performance describing in a system. Table 1 represents a summary of annotation for certain information (Merseguer and Campos, 2003). This study presents annotation to uncertain information for two behavioral diagrams of frequent usage among 5 behavioral diagrams (use case, sequence, collaboration, activity, state chart). To inspecting the other diagrams refer to (Ali, 2006).

It should be considered that the performance computations that are proposed fuzzily, are propounded for both type of information systems, which are extended with UML and F-UML (being fuzzy the performance computations is independent from the system development type).

Supporting performance concepts by an extended version from F-UML is introduced. This version uses a word named tag definition that has three types below: System load, routing rate and system usage.

Tagged value allows that required information could be attached to the model elements according to the tag definition format. In the other hand tagged value prepares this possibility to set annotation, which relates to the performance, on the model elements.

If tagged value is certainly, it will include data value from *string* type and else it will include data value from fuzzy type (the data value domain is fuzzy set). As an example the annotation with the name of message size that it is from system load type, has {100k} data value in certain information, though it has little data value in uncertain information.

Use case diagram: In this diagram actors, use cases and the relationships between them are inspected. The relationship between the actor and the use case is association, the relationship between the actors is generalization and the relationship among the use cases are generalization, include and extend. Among the propounded use cases in a diagram, some of them are important from performance aspect and they are inspected.

The role of use case diagram in performance evaluation process: By use cases diagram it can be specified that how much each actor uses the system. In Cortellessa and Mirandola (2000), a probability is allocated (the probability of executing the use cases by the actor) to each relationship between use case and actor. Supposing that there is a use case diagram with m users and n use cases and we have:

p_i (i = 1, ..., m) of i-th user usage from the system.
p_{ij} (j = 1, ..., n) the probability of using the i-th user from j-th use case

$$\sum_{j=1}^{n} P_{ij} = 1 \text{ and } \sum_{j=1}^{m} P_i = 1$$

Then the probability of executing the sequence diagram the x-th use case is Eq. to:

$$P(x) = \sum_{i=1}^{m} P_i . P_{ix} \quad (1)$$

Toward a fuzzy approach for computations related to performance in use case diagram: We use linguistic variables and fuzzy rules for entering uncertainty into the performance computations. For producing a fuzzy rules base, we will have a set of fuzzy if then rules like as under:

$$R^{(1)}_u: \text{IF } x_1 \text{ is } A^1_1 \text{ and}$$

$$x_2 \text{ is } A^1_2 \text{ Theny is } B^1$$

where:
A^l and B^l = In order to fuzzy sets in $U_i \subset R$ and $V \subset R$.
x = $(x_1, x_2)^T$.
ϵ U and y = In order to input and output variables (linguistic) in fuzzy system.

M is the number of existence rules in fuzzy rule base: l = 1, 2, ..., M.

The selected fuzzy system has product inference engine, singleton fuzzifier and center average defuzzifier:

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y} (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}{\sum_{l=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))} \quad (2)$$

where, B^l fuzzy set is a natural set with $\bar{y}^l$ center.

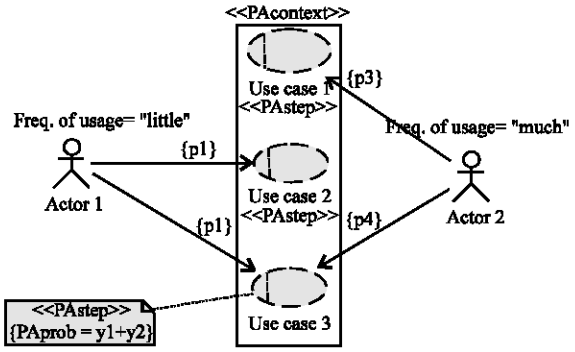

Fig. 5: Use case diagram with performance annotation (fuzzy approach)

Let us suppose to have the following fuzzy variables:

x_1 = The i-th user usage frequency from system.
x_2 = The probability of the i-th user usage from selected use case.
y = The probability of using the *i*-th user from the selected use case.

Then (for example) the following fuzzy rule can be propounded:

if x_1 is a little high and x_2 is high then y is high.

Defuzzifier is a mapping of B^l fuzzy set in $V \subset R$ (fuzzy inference engine output) to a $y^* \in V$certain point. This certain point, is the same probability of executing the use case and use it by the i-th actor. The Fig. 5 shows use case diagram with performance annotation (fuzzy approach). With supposing that 2 actors use the use case, in the same way, the probability will be computed for the second actor too. The sum of these two values is the probability of using the selected use case in the system. As it was said in the Table 1, system usage is counted as one of the system performance requirements.

In the Fig. 5 y_1 and y_2 are the outputs of fuzzy system and p_1, p_2, p_3 and are the fuzzy sets.

Sequence diagram: The preliminary descriptions about this diagram have described in the study. The use case diagram has 2 dimensions. The vertical dimension of diagram shows the time and the horizontal dimension of it shows the class instances (objects).

The role of sequence diagram in performance evaluation process: In this diagram, objects can be on the one system or different systems. In the centralized system, the spent time to dispatch a message is not important for performance modeling. Although, actions need some

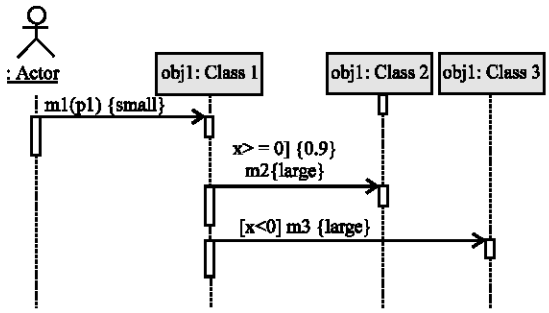

Fig. 6: Sequence diagram with performance annotation (fuzzy approach)

times to response to the message for computations, but investigation to this subject can be considered in state chart diagram (Ali, 2006; Merseguer and Campos, 2004). But for distributed systems, the messages, which are traveled through the net, allocate a noticeable time to themselves that can be propounded in system load.

In the hand we can attach a condition to each message in diagram and based on it, specify the probability of sending message. From the performance view point, this subject is leading to attach the routing rate on the message. For example, the messages m2 and m3 are showed in Fig. 6. In Bernarsi *et al*. (2002), tagged value is described for message size and the probability of message sending.

Toward a fuzzy approach for computations related to performance in sequence diagram: The response time in distributed systems depends on message size and net speed. To be aware of the message size and net speed as linguistic variables and fuzzy sets (like: large, medium and small) as linguistic variables domain, for each message we can compute the message sending time as the fuzzy system output. For example from fuzzy rules base, we can suppose the following fuzzy if then rule (the fuzzy system is expressed according formula 2):

If the message size is small and the net speed is high, then the response time is low.

Figure 6 shows the sequence diagram with performance annotation (fuzzy approach).

After drawing the F-UML diagrams with performance annotation, in the next step the diagrams are transformed to the fuzzy formal models (fuzzy Petri Net) and with usage of analysis techniques, we will solve the model (system analysis). For more Information about this subject you can refer to Ali (2006), Ali and Teshnehlab (2007) and Merseguer and Campos (2004).

IMPLEMENTATION A SUPPOSED SYSTEM AND SPECIFY THE ADVANTAGES OF THIS METHOD

As a case study, bank system is inspected to loan receive and deposit benefit payment. The loan receive manner in the mostly banks is in this way that 2 parameters are effective on loan amount specify: the account opening date and the balance amount. For example at least n monetary unit should be existed in the account for the specific time (in this case, one year) till loan would be paid to the customer. Now if the customer withdraws his money before the mentioned time, the loan will not belong to him. This subject is true in the deposits benefit computation to: If the customer has opened the one year account, he couldn't take his money before the appointed time. The crisp and certain essence of computations can be improved with the fuzzy sets. By presenting a fuzzy system according to formula two and by using the fuzzy rules, the amount of belonging loan to the customer is computed. An instance from fuzzy rule base:

If the account balance is little and the amount
has been in account for a low time then
the loan amount will be little.

The implemented system has been done by using the Rational Software, Matlab Tool (for fuzzy computation) and Pipe2 Tool (for generalized stochastic Petri Net implementation). Figure 7 represents the use case diagram and Fig. 2 represents the banking account for this system. In Ali (2006), in addition to fuzzy Petri Nets drawing, the following cases are specified too and necessary feedbacks for improving the functionality and performance are presented to the system:

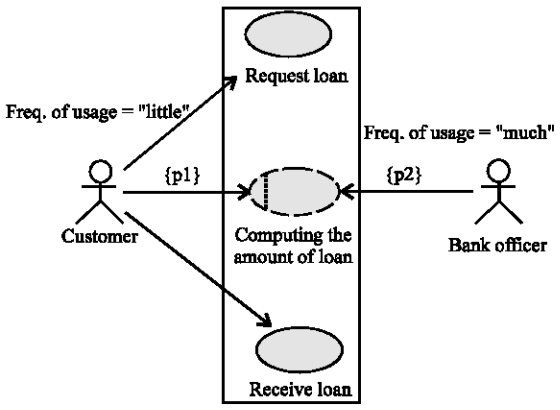

Fig. 7: Use case diagram with tagged value in a fuzzy approach for loan take system

- System uncontrollable states (from functionality aspect).
- The usage level from use cases and with help of it, specifying the system *utilization* (useful job in time unit).
- Different scenarios response time.

CONCLUSION AND FUTURE WORK

In this study, a novel method for development and evaluation of uncertain information systems was explained. The existence of uncertainty in natural problems is a common subject, that we used F-UML approach for investigation to it.
The advantages for this method are:

- Uncertainty entering in information systems modeling (from functionality and performance aspects).
- The usage of a standard language and exploitation from its advantages for consistency with software development different sections.
- Investigation to systems structure and behavior aspects.
- Defining profile for supporting the uncertainty without conflicting with UML standard profile (in system evaluation section).
- The ability of method executing on complex systems.

Because the idea is new and the usage is abundant, different researches can be defined in this context. Software methodologies that support uncertainty can be propounded as an instance from these researches.

REFERENCES

Ali, H. and M. Teshnehlab, 2007. Applying fuzzy-UML for uncertain systems modeling. First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran.

Ali, H., 2006. Design an Intelligent Model in Applying Fuzzy Object Oriented Databases for Systems Development, Ph.D. Proposal, Azad University, Science and Research Branch Tehran, Iran.

Bernardi, S., S. Donatelli and J. Merseguer, 2002. From UML Sequence Diagrams and State charts to analyzable Petri Net models. Proceedings of the Third International Workshop on Software and Performance, ACM Press, pp: 35-45.

Brooks, F.P., 1987. Essence and accidents of software engineering. IEEE Comput. 20 (4): 10-19.

Cortellessa, V. and R. Mirandola, 2000. Deriving a queueing network based performance model from UML diagrams. In: Proceedings of the Second International Workshop on Software and Performance, Ottawa, Canada, ACM, pp: 58-70.

IBM Company, 2003. Rational Software, http://www.rational.com, version.

Merseguer, J. and J. Campos, 2003. Exploring roles for the UML diagrams in software performance engineering, Proceedings of the 2003 International Conference on Software Engineering Research and Practice (SERP'03) (Las Vegas, Nevada, USA), CSREA Press, pp: 43-47.

Merseguer, J. and J. Campos, 2004. Software Performance Modeling using UML and Petri nets. Lecture Notes Comput. Sci. J., 2965: 265-289.

Object Management Group, 2002. UML Profile for Schedulabibity, Performance and Time Specification, http://www.omg.org.

Smith, C.U. and L.G. Williams, 2000. Software Performance anti Patterns, in Proceedings of the Second International Workshop on Software and Performance, Ottawa, Canada, ACM, pp: 127-136.

Smith, C.U., 1990. Performance engineering of software systems. The Sei Series in Soft-ware Engineering, Addison-Wesley.

Zongmin Ma, 2005. Fuzzy information modeling with the UML, Advanced in fuzzy object oriented databases: Modeling and applications, Idea Group Publishing.