

## Performance evaluation of enterprise architecture using fuzzy sequence diagram

Mohammad Atasheneh<sup>a\*</sup>, Ali Harounabadi<sup>b</sup> and Seyyed Javad Mirabedini<sup>b</sup>

<sup>a</sup>Department of Computer Engineering, science And Research Branch, Islamic Azad University, Khouzestan, Iran

<sup>b</sup>Faculty Member Of Computer Engineering, Central Branch, Islamic Azad University, Tehran, Iran

### CHRONICLE

#### Article history:

Received May 18, 2013

Accepted July 18, 2013

Available online

July 22 2013

#### Keywords:

Enterprise Architecture

Fuzzy UML

Fuzzy petri nets

C4ISR framework

Performance Evaluation

### ABSTRACT

Developing an Enterprise Architecture is a complex task and to control the complexity of the regulatory framework we need to measure the relative performance of one system against other available systems. On the other hand, enterprise architecture cannot be organized without the use of a logical structure. The framework provides a logical structure for classifying architectural output. Among the common architectural framework, the C4ISR framework and methodology of the product is one of the most popular techniques. In this paper, given the existing uncertainties in system development and information systems, a new version of UML called Fuzzy-UML is proposed for enterprise architecture development based on fuzzy Petri nets. In addition, the performance of the system is also evaluated based on Fuzzy sequence diagram.

© 2014 Growing Science Ltd. All rights reserved.

## 1. Introduction

Enterprise architecture process normally incorporates a kind of transition from the present state architecture towards the target state architecture (Shin et al., 2003). The process often consists of three phases of information technology, planning and execution and each component must be executed, consecutively. Measuring the performance of such architectures plays essential role in today's competitive industry. Mozaffari et al. (2011) implemented formal models and created an executable framework using Colored Petri Nets, which was used for validating the behavior and the performance of some architecture. There are presently some limitations on definitions of business process in the present literature and consequently their related models of business processes are suffering from some shortages. Lindsay et al. (2003) stated that most definitions were based on machine metaphor kind explorations of a process. While these methods are rich and illuminating they explained that they were too limited to state the true nature of business processes which is needed to develop and adapt to today's challenging environment.

\* Corresponding author.

E-mail addresses: [m\\_atasheneh@yahoo.com](mailto:m_atasheneh@yahoo.com) (M. Atasheneh)

The Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework, Version 2.0, developed by the U.S. Department of Defense (DoD) C4ISR Architecture Working Group, gives some guidance for explaining architectures (AWG, 1997). Afshani (2002) presented a new template based on fuzzy-UML concept based on fuzzy theory (Zadeh, 1983) for some of C4ISR products such as Logical Data Model (OV-7), Operational Event/Trace Description (OV-6c) and Systems Event/Trace Description (SV-10c). They also presented fictional Fast Pass system used at OilCo gas stations to demonstrate details of the proposed model. Bai (2008) presented an application with UML-based Petri nets for C4ISR Architecture simulation validation. Haroonabadi et al. (2008) investigated the behavioral description of uncertain information systems using fuzzy-state diagram. They also investigated the introduction of behavioral diagrams role in F-UML in software performance modeling process.

Computer applications in non-traditional area normally need conceptual data modeling and there have been many conceptual data models, being the tool of design databases, have been developed. The Unified Modeling Language (UML) is normally implemented for different areas of software and knowledge engineering, and it is being used to data modeling. There is also no doubt that information in real world application is often vague or ambiguous. Ma et al. (2011) presented different levels of fuzziness into the UML class diagram model and the corresponding formal representations were investigated based on fuzzy set and possibility distribution theory. They also extended the UML class diagram to fuzzy UML data model. Ma et al. (2012), in other works, introduced various levels of fuzziness into the class of UML and presented the corresponding graphical representations, with the result that UML class diagrams could model fuzzy information. The fuzzy UML data model was also formally mapped into the fuzzy object-oriented database model.

Finally, Bostan-Korpeoglu and Yazici (2004) explained that Fuzzy Petri Net (FPN) structure is used to represent knowledge and model the behavior in many intelligent object-oriented database environment, which integrates fuzzy, active and deductive rules with database objects. Nevertheless, the behavior of a system could be unpredictable because of the rules triggering each other. Intermediate and final database states could also differ depending on the order of rule executions. In order to foresee and handle problematic behavior patterns, they employ a static analysis on the FPN structure, which provided easy checking of the termination property without requesting any extra construct. Besides, with the proposed fuzzy inference algorithm, they make sure about confluent rule executions.

## **2. Data modeling, fuzzy class and sequence diagram**

As mentioned in the previous section, Enterprise Architecture Framework C4ISR has many products and there have been only few applications where the performance of this system was effectively measured. These products include: SV-4, SV-10, SV-11. Each of these products were used for their expression of one or more of the UML diagrams. To model the product SV-4, which describes the operation of the systems from activity diagrams, we may use state diagrams and sequential product of SV-10 and SV-11 of the class diagrams. So initially, we may apply these figures with linguistic variables and the corresponding phase diagrams can be converted.

### *2.1. Fuzzy UML data model*

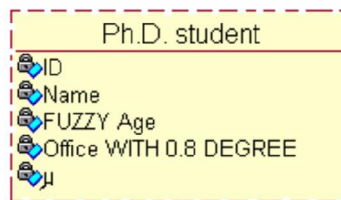
In modeling fuzzy data, UML diagrams are implemented and UML class diagrams are logical models for expressing the basic structure of the system. Classes and relationships among different components are the constituent elements of a class diagram. When there are uncertainties associated with these elements, we may create Fuzzy-UML data model. According to Ma (2004, 2005) and Ma et al. (2011, 2012), there are three levels of fuzzy optimization as follows,

1. Construction phase where the contents of the class in the data model as the fuzzy characteristics is created.

2. Construction phase where some of the objects may be members of a class based on a membership function.
3. The third phase of the process, the attribute value is typical of the class. An attribute in a class, on the assumption that the domain of values to be defined.

Attribute or class name, as appropriate, with the words “*WITH MEM DEGREE*”, where  $0 \leq MEM \leq 1$  is to be expressed. This value demonstrates the degree in which an attribute belongs to a given class or classes of models. At the third level, a keyword fuzzy character appears. In the second phase, we define the degree that must be awarded to an instance of a class. For this purpose, an additional attribute in class shows the membership, denoted as  $\mu$ , of a sample to a class, which is defined in the range of  $[0,1]$ . Fig. 1 shows a class of fuzzy Ph.D student. The class attribute age can be a little fuzzy. PhD students can be in office or not. However, a PhD student could be out of his/her office for a probability of about 80%. In this property office, a PhD student from the class is uncertain. This class belongs to the first level of fuzzy expression for the fuzzy definition.

“*WITH 0.8 membership DEGREE*”



**Fig. 1.** Representation of a fuzzy class

## 2.2. Sequence diagram of fuzzy logic to fuzzy user realizes

To determine the behavior of the system, functional requirements must first be determined. Classified extraction software engineering approach must be stated based on four characteristics including objectives, process, and service functions. In the object-oriented approach, services are important to us, and not the exchanged information. Each service user is defined in terms of a call. When a service is under an uncertain expression, the use phase is discussed. For each user, we may provide the service scenario, which means the user has to be adjusted. If the user is considered as an indeterminate form, a sequence diagram is also uncertain.

## 3. Fuzzy Petri Net

According to Bostan-Korpeoglu and Yazici (2004, 2007) and Motameni et al. (2008), Fuzzy Petri nets for modeling fuzzy rules can be defined as follows,

$$(P, P_s, P_e, T, TF, TRTF, A, I, O, TT, TTF, AEF, PR, PPM, TV), \quad (1)$$

where  $P$  is a finite set of places and other parameters are defined as follows,

$P_s \subset P$ :	A finite set of input events is the primary location.
$P_e \subset P$ :	A set of output locations for operations or results
$T$ :	This is a finite set of phase transitions. The transition from the values of input fields and values are applied to place the output they produce.
$TF: T \rightarrow TF$	T is the transition function of each member to a member function mapping to TF.
$A \subset (P * T \cup T * P)$	Is a finite set of connections between places and transitions. Input connections between places and transitions ( $P * T$ ) and the connections between investment and output spaces ( $T * P$ ) are expressed by them.

$I = P \rightarrow T$	Is an input mapping.
$O = T \rightarrow P$	A mapping of the output.
$TT$	A finite set of token types is fuzzy. Each token is a linguistic variable (eg "Low, Medium), which are expressed by a membership function.
$TTF := T \rightarrow P$	Each token type is a fuzzy member function where $P$ is mapped to a $TT$ member token phase.
$AEF : Arc \rightarrow Expression$	Is a function that maps each connected to a string that will hold the information.
$PR :$	Is a finite set of propositions that correspond to an event or condition or operations.
$PPM : P \rightarrow [0,1]$	A fuzzy space is mapped statement $  PR   =   P  $ .
$TV : P \rightarrow [0,1]$	$\mu_i$ is the value of tokens that can be mapped to locations that Degree of membership of a token to a specific place, says.

This paper presents an algorithm to evaluate enterprise architecture, which helps executives and decision makers make appropriate decisions in enterprise architecture. It also helps to articulate some properties of linguistic variables to combine fuzzy logic for evaluating enterprise architecture. Note that UML is unable to provide a true model (UML) as formal fuzzy Petri nets models for software systems requirements. Therefore, we describe one product (sv-10), which uses the fuzzy graph sequence and present an algorithm that converts the phase diagram into colored Petri nets (Motameni et al., 2008). The algorithm is as follows:

Step 1) For each message in sequence diagram, all events and conditions should be determined.

Step 2) The right conditions must be investigated. Therefore, we should be mapping it. For each condition, a responsible transition is in charge of validating the results. This means that the token is expressed with a fuzzy value between 0 and 1. In most cases, we deal with more than one condition but here this condition can be expressed as follows,

$$OR = \mu_{AB}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2)$$

$$AND = \mu_{AB}(x) = \min[\mu_A(x), \mu_B(x)] \quad (3)$$

Step 3) Run events, until the conditions are established. The operation uses fuzzy values calculated for the assessment and transition is applied prior to the application of the membership function. Finally, given the final output of a fuzzy system, a value is determined. The operation is performed based on COG test as follows,

$$COG = \frac{\int_a^b \mu_A(x) dx}{\int_a^b \mu(A) dx} \quad (4)$$

#### 4. Case study

In this section, we apply the proposed algorithm on an automatic washing machine. The washing machine does its operations based on the degree of dirt and grease. Thus, task allocation method is responsible for the dirty clothes and fat and fuzzy method should be able to dress a degree between 0 and 1, to give them the dirt (of the second phase). Fig. 2 shows the phase diagram washing system:

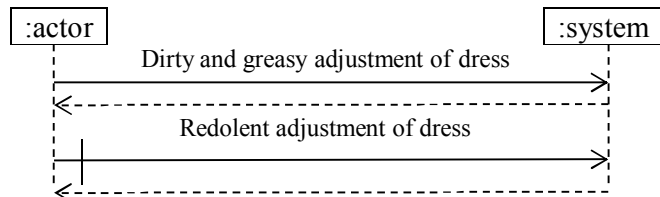


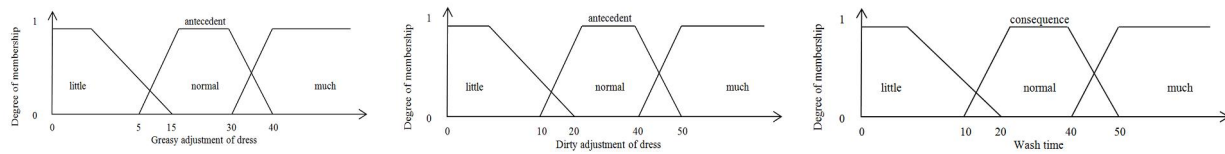
Fig. 2. Fuzzy sequence diagram for washing System

To estimate the response time of the system, first the necessary laws are stated and then fuzzy events are expressed in Table 1.

**Table 1**  
Rules And Scenarios Washing System

Rule	Event	Condition	State
$R_1$	Dress is Dirty	If (Dress is Much Dirty) the	Wash time is min
$R_2$	Dress is Greasy	If (Dress is Little Greasy) then	Wash time is Medium
$R_3$	Dress is Dirty And Dress is greasy	If (Dress is Much Dirty) and If (Dress is Much Greasy) then	Wash time is Long

In addition, Fig. 3 shows details of membership functions,



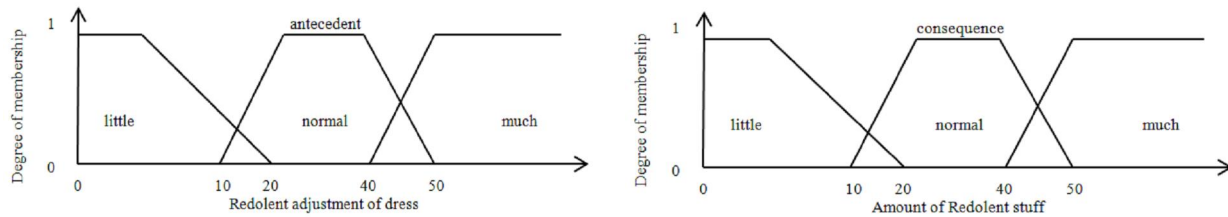
**Fig. 3.** Membership functions

The first method of input are associated with linguistic variables and fuzzy terms, but the method is not fuzzy. Therefore, a second method called fuzzy redolent of the method is to be added to the system. Table 2 shows the rules used to redolent the method:

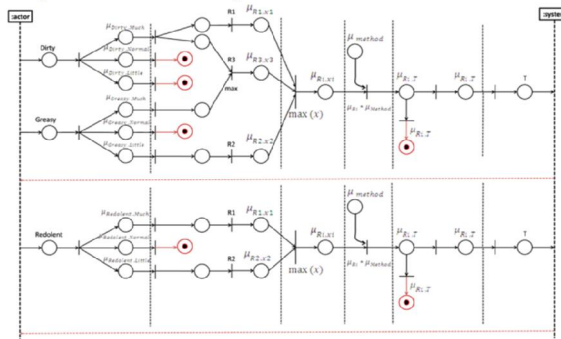
**Table 2**  
Terms and conditions of redolent method

Rule	Event	Condition	State
$R_1$	Dress is redolent	If (much redolent is required) then	Set redolent to many
$R_2$	Dress is redolent	If (little redolent is required) then	Set redolent to low

In addition, Fig. 4 and Fig. 5 show details of membership functions for redolent method and the results of the implementation of our proposed method, respectively.



**Fig. 4.** Membership function redolent method



**Fig. 5.** The results of the implementation of the proposed method

In this section, we analyze the performance model of the architecture, which would deal with stakeholder viewpoints. To simulate the Petri graph, the software CPN / TOOLS AND SHOW

FLOW were used. Response time is the time when a person or organization uses the services of an agent for a particular service. Here, we assume that the arrival time of the system is exponentially distributed with parameter 0.1, In addition, the service time for each customer for each server is exponentially distributed with parameter 0.5 and there are 2 servers. In addition, the departure time of the system is exponentially distributed with parameter 0.3. In our survey, the number of jobs processed by server 1 and server 2 are 1028 and 927, respectively. In addition, the response times for two servers are 12.2356 and 11.5348, respectively.

## 5. Conclusion

In this paper, we have presented an algorithm for evaluating the performance of C4ISR enterprise architecture framework. The proposed method first intended to use Fuzzy-UML for performance evaluation. However, since UML was unable to evaluate software systems, the proposed algorithm transformed the corresponding phase diagrams into Petri nets. Since most of the actual system requirements are inconsistent with the proposed model to cover this issue, we used fuzzy logic. Given that the majority of software systems, specific parameters were used to evaluate performance of any system.

## References

- Afshani, J., Harounabadi, A., & Dezfouli, M. (2002). A new model for designing uncertain enterprise architecture. *Management science letters*, 2(2), 689-696.
- Bai, X. H. (2008, July). An application with UML object-based PETRI NETS for C4ISR architecture simulation validation. In *Machine Learning and Cybernetics, 2008 International Conference on* (Vol. 4, pp. 2257-2263). IEEE.
- Bostan-Korpeoglu, B., & Yazici, A. (2004). Using fuzzy petri nets for static analysis of rule-bases. In *Computer and Information Sciences-ISCIS 2004* (pp. 72-81). Springer Berlin Heidelberg.
- Bostan-Korpeoglu, B., & Yazici, A. (2007). A fuzzy Petri net model for intelligent databases. *Data & Knowledge Engineering*, 62(2), 219-247.
- Haroonabadi, A., Teshnehlab, M., & Movaghar, A. (2008). A novel method for behavior modeling in uncertain information systems. *World Academy of Science, Engineering and Technology*, 41, 959-966.
- Lindsay, A., Downs, D., & Lunn, K. (2003). Business processes—attempts to find a definition. *Information and Software Technology*, 45(15), 1015-1019.
- Ma, Z. (2004). *Extending UML for Fuzzy Information Modeling in Object Oriented Database*. Theories and Practices, Idea Group Publishing.
- Ma, Z. (2005). *Fuzzy Information Modeling With the UML*. Idea Group Publishing, 153-176.
- Ma, Z. M., Zhang, F., & Yan, L. (2011). Fuzzy information modeling in UML class diagram and relational database models. *Applied Soft Computing*, 11(6), 4236-4245.
- Ma, Z. M., Yan, L., & Zhang, F. (2012). Modeling fuzzy information in UML class diagrams and object-oriented database models. *Fuzzy Sets and Systems*, 186(1), 26-46.
- Motameni, H., Movaghar, A., Daneshfar, I., Nemat Zadeh, H., & Bakhshi, J. (2008). Mapping to convert activity diagram in Fuzzy UML to fuzzy Petri Net. *World Applied Sciences Journal*, 3(3), 514-521.
- Mozaffari, M., Harounabadi, A., & Mirabedini, S.J. (2011). A method for validating a behavior of enterprise architecture. *World Applied Sciences Journal*, 14(6), 831-841.
- Shin, M. E., Levis, A. H., & Wagenhals, L. W. (2003, October). Transformation of UML-based system model to design/CPN model for validating system behavior. In *Proc. of the 6th Int. Conf. on the UML/Workshop on Compositional Verification of the UML Models*.
- US DoD C4ISR Architecture Working Group (AWG) (1997). *C4ISR Architecture Framework Version 2.0*. Washington, D.C.
- Zadeh, L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy sets and Systems*, 11(1), 197-198.