ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Software Requirements Modeling From a Selected Set of Requirements Using Fuzzy Based Approach

Mohd. Arif[1], Chaudhary Wali Mohammad[2] and Mohd. Sadiq[3]

ABSTRACT

Software requirements (SRs) modeling is one of the sub-processes of requirements engineering whose aim is to model and design the SRs before the development of a project. Different techniques are employed for representing the SRs using goal concepts, unified modeling language, etc. There could be a large number of SRs after the completion of the requirements elicitation process. It is not practical to model the complete set of the identified SRs because of the cost, time, and other limitations of an organization. There should be some systematic methodology to identify and select those SRs for modeling that need to be implemented during the software development process. The selection of SRs from the list of the elicited requirements is a multi-criteria decision-making process. In this process different stakeholders participate in the selection of the SRs. Linguistic variables may be used by the stakeholders to specify the preferences of SRs. To deal with this issue, a method has been proposed using a fuzzy based approach so that the selected set of SRs can be modeled and implemented during the development phase. The proposed method is explained by considering the small and large set of SRs for an institute examination system. The ranking value of the functional requirements of an examination system is computed. Based on the ranking order, top three requirements are modeled using use-case diagrams (UCDs) and class diagrams. It is found that both diagrams represent different information about the requirements of an examination system and there is no overlap in the information captured through UCDs and class diagrams.

Article information:

Keywords: Functional Requirements, Non-functional Requirements, Requirements Modeling, Goal Oriented Modeling, Class Diagram and Use-case Diagram

Article history:
Received: December 30, 2021
Revised: February 5, 2022
Accepted: March 4, 2022
Published: June 4, 2022 (Online)

DOI: 10.37936/ecti-cit.2022162.247272

1. INTRODUCTION

The stakeholders are the key source of the software requirements (SRs) which are identified by requirements elicitation techniques. These requirements are represented by different modeling techniques like goal-based methods, unified modeling language (UML), etc. [1]. To represent the SRs according to the needs of the project, it is indispensable to elicit and select the SRs before the beginning of the modeling process. Requirement's modeling is one of the key sub-processes of requirements engineering which helps people model and understand the requirements using different methods.

It also helps to identify the dependencies among the SRs. [2]. It has been noticed that the needs and understanding of stakeholders about the requirements changes continuously. As a result, this may lead to an incomplete set of requirements. Under this situation it is difficult to model the SRs correctly. Hence, SRs should be identified according to the consensus of the stakeholders using the requirements elicitation techniques so that requirements can be modeled properly before starting the implementation of software.

UML diagrams have been classified into structural and behavioral diagrams as shown in Figure 1. These diagrams are suitable only for modeling the function-

[1,2]The authors are with the Computer Science and Technology Research Group, Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia, A Central University, New Delhi, India, E-mail: mohd.arif.jmi@gmail.com and cmohammad@jmi.ac.in
[3]The author is with the Software Engineering Laboratory, Computer Engineering Section, UPFET, Jamia Millia Islamia, A Central University, New Delhi, India, E-mail: msadiq@jmi.ac.in, sadiq.jmi@gmail.com
Corresponding author: msadiq@jmi.ac.in, sadiq.jmi@gmail.com

ality of a system [3, 4, 5].

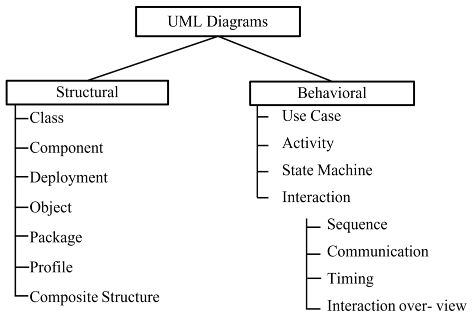


Fig. 1: *Classification of UML diagrams.*

In real life applications, a system may have several SRs and it is not possible to model all of the elicited requirements because of budget, time, and other constraints of an organizations. This motivated us to develop a methodology for modeling the selected set of requirements of a system so that a successful system can be developed. This type of methodology will be useful for those people who are working in the software industry around the world.

One of the benefits of modeling the selected set of requirements is identifying those requirements that will be modeled and implemented during the different phases of software development. Such a system will be useful in software industries because during the development process several stakeholders participate and capture their opinions during the modeling and analysis of the SRs, and that plays an important role for developing a system according to the consensus of the stakeholders.

Different studies have investigated UML diagrams in the context of SR representation [6, 7]. For example, Siau and Lee [7] examined the importance of UCDs to interpret the requirements when UCDs are used in conjunction with a class diagram. Research in the area of SRs modeling has focused on the following areas: requirements traceability, reusability, verification of UML models, generating UML diagrams from requirements, integrating Z notations in UML diagram, etc. Most of the work in the area of SRs modeling is based on how to apply the different modeling techniques for representing the SRs. Less attention is given to how to model those SRs which are more important during the development of software. There should be some systematic methodology to select those requirements for modeling which are more important during the development process. To address this issue, in [5] we proposed a method for modeling the selected set of SRs in which the class diagrams were used for modeling the requirements of the examination system of an educational institute. In [5], the requirements were selected for modeling and analysis by using the fuzzy "analytic hierarchy process" (AHP).

Uraipan et al. [8] applied the fuzzy AHP method for the selection of the level of cyber resilient capability maturity model. The fuzzy AHP is limited to the number of alternatives or "*functional requirements*" (FRs) and criteria or "*non-functional requirements*" (NFRs). Fuzzy AHP is a better choice for smaller numbers of FRs and NFRs. It cannot handle a large set of requirements [9]. Therefore, to deal with large sets of SRs, this paper presents a method for modeling a selected set of SRs in which the "*Technique for Order of Preference by Similarity to Ideal Solutions*" (TOPSIS) has been applied under fuzzy environment [9] to elicit the ranking values of the requirements so that a large set of SRs can be modeled and analyzed. The contributions of our work are given below:

1. A method has been developed for modeling a selected large set of SRs.
2. A program has been written using Visual C++ (VC++) language to identify the ranking order of the SRs using fuzzy TOPSIS.
3. Both UCDs and class diagrams have been employed for modeling and analysis of the structural and behavioral aspects of an IES.

This paper is structured as follows: The related work on modeling the SRs is discussed in Section 2. The proposed method for modeling the selected set of SRs is presented in Section 3. A case study is given in Section 4. The comparative study between the proposed method and other selected methods is discussed in Section 5. Finally, the conclusion and future work are discussed in Section 6.

2. RELATED WORK

An insight into UCDs and class diagrams is discussed in Section 2.1. Different methods for modeling the SRs are presented in Section 2.2.

2.1 Class diagram and use-case diagram

Class diagrams are used to model the static view of object-oriented systems because they are concerned with the classes, interfaces, associations, and constraints. The class diagrams are used for expressing, visualizing, and documenting the SRs of a system. In Figure 2, we have modeled some requirements of an Institute Examination System (IES) using a class diagram. Relationships among the classes are represented by lines. Broken arrows indicate the services of one class used by another class.

In Figure 2 *Student Class* is inherited by both the *External Students* and *Internal Students* classes. On the other hand, the *Teacher* class is evaluating both *External Students* and *Internal students* to produce the marks which are stored in the *Result* class [4].

The use-case diagrams (UCDs) capture the functional aspects of the system. They represent the sequence of activities when an actor interacts with the system. The UCDs are the graphical representation of the SRs which can be decomposed and refined into different levels of abstraction [5]. The UCDs describe the sequence of interactions between actors and system so that required services can be provided to the end users.

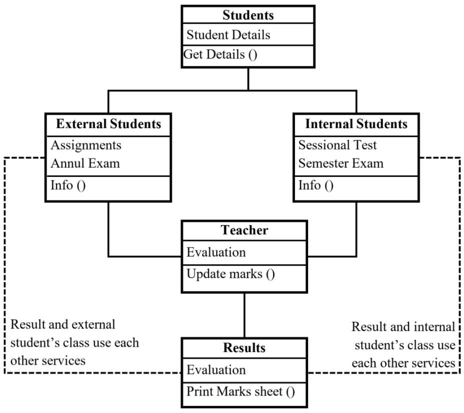


Fig.2: *Class diagram for some requirements of an IES.*

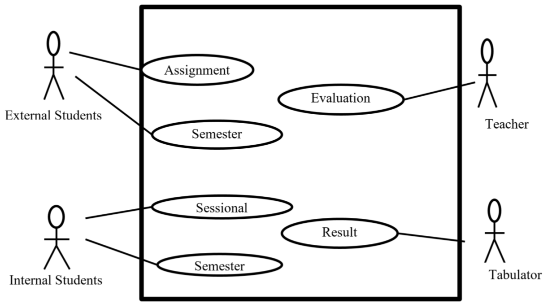


Fig.3: *UCD for some requirements of an IES.*

A UCD has various components, such as actor, use-case, and their relationships. Figure 3 represents the UCD in which internal and external students are interacting with the system to get the results of the semester. Here, we have four types of actors who are interacting with the system. The aim of teacher actor is to enter the marks of both internal and external students after evaluating answer sheets so that the tabulator can tabulate the marks which are given by the concerned teachers. It is clear from both Figure 2 and Figure 3 that class diagrams and UCDs play an important role in understanding the requirements so that a successful product can be developed [7]. This motivates us to apply both the UCD and class diagrams for the modeling and analysis of the SRs.

2.2 Methods for modeling the software requirements

There are various possible methods to model and analyze the SRs using UML diagrams [4, 5]. To strengthen the early phase of the software development lifecycle (SDLC), Chanda *et al.* [10] developed a method using formal grammar for representing the use-case, activity, and class diagrams. Verification of the requirements in their traceability in early phase of the SDLC may reduce the time and cost of the software product. In UML, structural aspects of design are modeled by class diagrams. The redundancies and multiple inheritances should be eliminated from the class diagrams so that the quality of the class diagrams can be improved. To remove the clones of attributes, Lano and Rahimi [11] presented a case study based on class diagram refactoring to improve the quality of the design of the class diagrams. To automate the task of the modeling of SRs, various methods have been developed. For example, Narawita and Vidanage [12] used "*natural language processing*" to generate the UML diagrams form the textual requirements so that the system analyst can save time.

Mohammad *et al.* [13] developed a method to extract information from use cases and translate it into class diagrams. The main objective of this method was to identify different classes and their relationships. Alsaadi [14] described an approach for measuring software performance using UML class diagrams. The performance values are predicted by use-cases and data associated with them. Among various notations in requirements modeling, Z notations have been used for the analysis of the system. This notation has been used in [15] for modeling the static and dynamic viewpoints of a system. To speed up the development process, the UML diagrams have been reused because constructing a UML diagram from the beginning is a time-consuming task. In literature, diagram similarity measure has been employed to identify reusable diagrams. For example, Fauzan *et al.* [16] developed a method which focused on identifying the similarity between two diagrams produced by structural and lexical information. In their work, the structural similarity was measured using the relationship between actors and use-cases.

On the other hand, the cosine similarity and some other methods have been employed to measure lexical information. Javed *et al.* [17] discussed a grammar-based approach for validating the class diagrams. In this approach a string comparison metric was employed for feedback so that class diagrams can be modified based on the suggestions of the stakeholders. The UML models should be simple because it is difficult to understand complex UML models. Several diagrams are constructed for the same system at various levels of abstraction so that consistency among the UML models can be achieved. There should be some set of rules for ensuring the consistency of UML diagrams at different levels of abstraction. To deal with this issue, Faitelson and Tyszberowicz [18] defined some set of rules which is referred to as diagram refinement. Thakur and Gupta [19] developed a tool supported method for generating the sequence diagrams from use cases which are written in natural language. The authors have used three case studies to show the applicability of the automated approach.

In literature, class diagrams have been considered as key diagrams for modeling the FRs of software. It is a challenging task to extract the class diagrams from natural language requirements. Karaa *et al.* [1] developed a tool to uproot the class diagrams from the SR. These requirements were written in natural language.

The existing methods in SRs modeling are dedicated to representing the requirements using UML models, generating different types of diagrams from the textual requirements, etc. [3]. A system may have a hundred or more requirements in the early phase of software development. It is not possible to model and implement all of the elicited requirements due to different constraints of an organization [20]. Selection of SRs for modeling from the set of the requirements is a "*multi-criteria decision-making*" (MCDM) process. In this process, different stakeholders participate and they may use linguistic variables to specify their preferences about the requirements. To deal with this issue, a method has been proposed using a fuzzy based approach so that the selected set of requirements can be modeled and implemented during the development phase. To model the linguistic variables, in our work, we have used "*triangular fuzzy numbers*" (TFNs) because of its simplicity [21]. To the best of our knowledge, there is no study in the literature which models the selected set of SRs using UML diagrams under a fuzzy environment. Therefore, in this paper an attempt has been made to select SRs for modeling which have a high priority.

3. PROPOSED METHOD

A block diagram of the proposed method is exhibited in Figure 4. The steps of the proposed method are given below:

- **Step 1:** Identification of stakeholders and goals of an organization
- **Step 2:** Elicitation of SRs using a goal-oriented approach
- **Step 3:** Evaluation of SRs by decision makers
- **Step 4:** Computation of the ranking values of FRs using fuzzy TOPSIS
- **Step 5:** Modeling of the selected set of FRs using use-case and class diagrams

Step 1: Identification of stakeholders and goals of an organization

The aim of this step is to identify different stakeholders and the goals of an organization. The stakeholders are the group of people or organizations who are interested in the development of a project. In this work, the identified stakeholders have been classified on the basis of influence (Inf) and interest ($Intr$) in the project: (i) when both Inf and Intr are high, (ii) when Inf is high and Intr is low, (iii) when Intr is high and Inf is low, (iv) when both Inf and $Intr$ are low [21]. Let $S = \{S_1, S_2, \ldots, S_n\}$ be the set of stakeholders who will participate during the elicitation of the SRs

Step 2: Elicitation of SRs using a goal-oriented approach

The goal oriented approach has been used for the elicitation of the SRs by the identified stakeholders. The objective of a goal oriented approach is to decompose and refine the goal (G) so that "*functional requirements*" (FRs) and "*non-functional requirements*" (NFRs) of a system can be identified from G. In this approach, G is decomposed and refined into sub-goals (SGs) until the responsibility for each SGs is assigned to some agent or system. These SGs are connected by "*logical AND*" and "*logical OR*" connectives and from an AND/OR graph. In this graph, suppose a SG "t" is connected to three SGs t_1, t_2, and t_3 using *logical AND* connectives. The *SG* "t" will be achieved only when all three connected SGs have been achieved. In the case of a *logical OR* connective, the achievement of any SG may lead to the achievement of SG "t" [25].

Step 3: Evaluation of SRs by decision makers

Let $p = \{DM_1, DM_2, \ldots, DM_p\}$ be the set of decision makers for evaluating the elicited SRs. In a real life application, the decision makers may use linguistic variables for the evaluation of the SRs [22, 23, 24]. These variables are modeled by different types of fuzzy numbers like "*triangular fuzzy numbers*" (TFNs), trapezoidal fuzzy numbers, sigmoidal fuzzy numbers, pentagonal fuzzy numbers, etc. Among these numbers, in our work, TFNs were chosen to model the linguistic variables during the computational process because of their simplicity. This makes them easy to understand in computation as well as in representation.

Step 4: Computation of the ranking values of FRs using fuzzy TOPSIS

The elicited FRs, i.e., $FR_1, FR_2, \ldots, FR_y$, are evaluated based on the NFRs, i.e., $NFR_1, NFR_2, \ldots, NFR_z$, by P decision makers (Dem) to construct the "*fuzzy decision matrix*" ($FDMat$). The fuzzy rating of each decision maker $Dem_p, p = 1, 2, \ldots, P$ can be represented using TFNs as $T_p = (r_p, s_p, t_p)$ $p = 1, 2, \ldots, P$ with membership function $\mu_T(x)$. After that the fuzzy rating is aggregated and it can be calculated as $T = (r, s, t), p = 1, 2, \ldots, P$. Here,

$$r = \min\{r_p\}, s = \frac{1}{p}\sum\nolimits_{p=1}^{P} s_p, \text{ and } t = \max\{t_p\}. \quad (1)$$

Suppose the fuzzy rating of the p^{th} DeM for a FR is $k_{nmp} = (r_{nmp}, s_{nmp}, t_{nmp})$, where $n = 1, 2, \ldots, y$, and $m = 1, 2, \ldots, z$. Here, y and z are the number of FRs and NFRs, respectively. The aggregated fuzzy rating (k_{nm}) of FRs with respect to NFRs can be

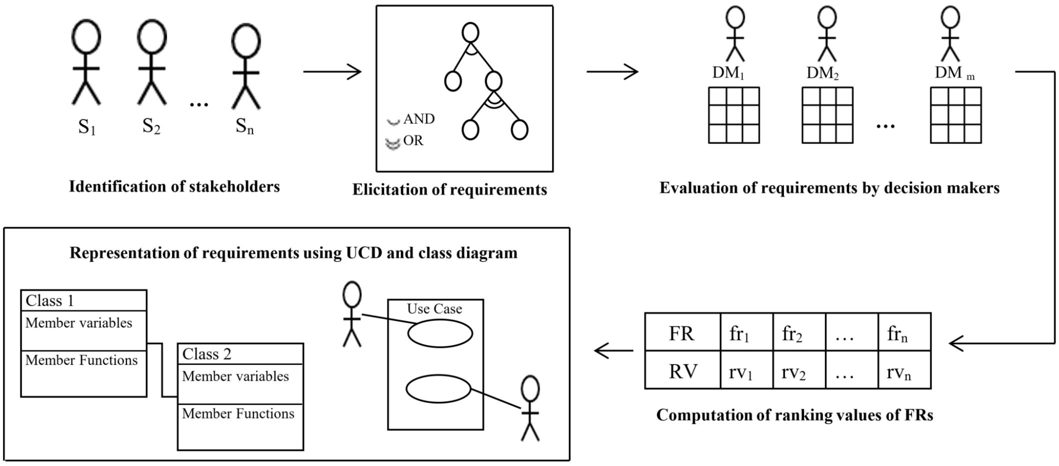


Fig.4: *Block diagram of the proposed method.*

calculated as $(k_{nm}) = (r_{nm}, s_{nm}, t_{nm})$. Here,

$$\mathrm{r}_{nm} = \min\{r_{nmp}\}, s_{nm} = \frac{1}{p}\sum_{p=1}^{P} s_{nmp}, t_{nm} = \max\{t_{nmp}\}. \quad (2)$$

Let the importance weight of an NFR by the p^{th} DeM be $wt_{nmp} = (t_{mp1}, t_{mp2}, t_{mp3})$, then the aggregated fuzzy weights (wt_{nm}) of each NFR is computed as:

$$wt_m = (wt_{m1}, wt_{m2}, wt_{m3}).$$

where

$$\mathrm{wt}_{m1} = \min\{r_{mp1}\}, wt_{m2} = \frac{1}{P}\sum_{p=1}^{P} s_{mp2}, wt_{m3} = \max\{t_{mp3}\}. \quad (3)$$

The $FDMat$ is constructed as:

$$FDMat = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1z} \\ k_{21} & k_{22} & \cdots & k_{2z} \\ \vdots & \vdots & \cdots & \vdots \\ k_{y1} & k_{p2} & \cdots & k_{yz} \end{bmatrix}$$

$$WT = [wt_1, wt_2, \ldots . wt_y]$$

Here, $k_{nm} = (r_{nm}, s_{nm}, t_{nm})$ and $wt_m = (wt_{m1}, wt_{m2}, wt_{m3})$; $n = 1, 2, \ldots, y$; $m = 1, 2, \ldots, z$ can be approximated by TFNs.

A linear scale transform has been used to normalize the $FDmat$ and we call it normalized $FDmat$, i.e., $NormFDMat$.

$$NormFDMat = [\mathrm{g}_{nm}]_{y\times z} \; n = 1, 2, \ldots, y \text{ and } m = 1, 2, \ldots, z \quad (4)$$

where

$$g_{nm} = \left\{\frac{r_{nm}}{t_m^*}, \frac{s_{nm}}{t_m^*}, \frac{t_{nm}}{t_m^*}\right\}$$

$$t_m^* = \max\ t_{nm}.$$

The weights of the NFRs are multiplied by the $NormFDMat$ to obtain the weighted $NormFDMat$ (V). It can be defined as:

$$V = [v_{nm}]_{y\times z}, n = 1, 2, \ldots, y \text{ and } m = 1, 2, \ldots, z \quad (5)$$

$$v_{nm} = g_{nm}(.)wt_m$$

Here, wt_m represents the importance weight of each NFR. Based on the weighted NormFDMat (V), the values of the "*fuzzy positive ideal solutions*" $(FPISol)$ and "*fuzzy negative ideal solutions*" $(FNISol)$ are calculated as:

$$FPISol = (FPISol_1, FPISol_2, \ldots, FPISol_z) \quad (6)$$
$$FNISol = (FNISol_1, FNISol_2, \ldots, FNISol_z) \quad (7)$$

where

$$FPISol_m = \max\{FPISol_{nm3}\}$$

$$FNISol_m = \min\{FNISol_{nm1}\}$$

$n = 1, 2, \ldots, y$ and $m = 1, 2, \ldots, z$

The values of $FPISol$ and $FNISol$ are used to compute the closeness coefficient $(ClosCoeff)$ values. The ranking order of the FRs is determined by the $ClosCoeff$. It can be calculated as:

$$ClosCoeff_n = \frac{d_n^{FNIDol}}{d_n^{FPISol} + d_n^{FNISol}}, n = 1, 2, \ldots, y \quad (8)$$

where

$$d_n^{FPISol} = \sum_{m=1}^{z} (v_{nm}, FPISol_m) n = 1, 2, \ldots, y \quad (9)$$
$$d_n^{FNISol} = \sum_{m=1}^{z} (v_{nm}, FNISol_m) n = 1, 2, \ldots, y \quad (10)$$

$d(.,.)$ is the distance measurement between two fuzzy numbers.

Step 5: Modeling of selected set of FRs using use-case and class diagrams The aim of this step is to select the FRs based on their priority value. The FRs having the highest ranking values will be used for modeling using UCDs and class diagrams.

4. CASE STUDY RESEARCH PROCESS

In this step the guidelines proposed by Runeson et al. [26] are adopted for conducting the case study research process (CSRP). This process includes the following sub-processes: (a) "*case-study design and planning*", (b) "*data preparation and collection*", and (c) "*data analysis*". These guidelines have already been adopted in different studies for conducting the CSRP. For example, Sadiq [21] adopted the guidelines of [26] for the identification and analysis of the stakeholders under a fuzzy environment. The explanation of the sub-processes of the CSRP is given next:

4.1 Case study design and planning

The aim of the "*case study design and planning*" (CSDP) step is to explain the following elements: (4.1.1) objective, (4.1.2) the case, and (4.1.3) research questions (RQs).

4.1.1 Objective

The objective of this study is to model the selected set of FRs of an Institute Examination System (IES).

4.1.2 The case

The second element of CSDP is concerned with "*what is studied?*" In this study, we are studying the IES so that different types of the activities related to an examination system can be implemented after modeling. The key activities related to an IES include the following [20, 21]: (a) download the hall ticket, (b) submit examination form, (c) download end semester mark-sheet of various courses offered by a university or an institution, etc.

4.1.3 Research questions

The following RQs have been formulated as a part of the CSDP sub-process:

- RQ 1: Which set of requirements should be modeled during the SRs analysis?
- RQ 2: Do the UCD and class diagrams capture the same information about the SRs during the requirements analysis?

4.2 Data preparation and collection

The FRs are evaluated on the basis of NFRs so that the opinions of decision makers can be collected for further analysis. In our work, the data has been collected after evaluating the FRs by five DMs based on the different NFRs. A description of the preparation and collection of the data is discussed in the next section.

4.3 Data analysis

The data was collected after evaluating the FRs and NFRs by the decision makers for the analysis. This data has been used for analyzing the formulated RQs and it is discussed in Sub-section 4.3.4. Here, we have considered both small and large datasets of the requirements of an IES. The large datasets have been adopted for the analysis from the work of Sadiq and Devi [27].

4.3.1 Step 1: Identification of stakeholders and goals of an organization

In our work, ten stakeholders have been identified, $S_1, S_2, \ldots, S_{10}$, for eliciting the requirements of a goal. Here, the director of an institute (S_1) and in-charge of the system development (S_2) have high Inf and high $Intr$ in the development process. The goal of S_1 is to automate the activities of the Controller of Examinations department. A list of the remaining stakeholders is given below:

- S_3: Financer or funding agencies
- S_4, S_5, and S_6: Team of requirements analysts for eliciting the FRs and NFRs of an IES
- S_7 and S_8: Team of developers
- S_9: Consultant
- S_{10}: Students or end users

4.3.2 Step 2: Elicitation of SRs using a goal-oriented approach

The aim of this step is to elicit the requirements according to the needs of the stakeholders. The "*goal-oriented approach*" has been applied to elicit the SRs of an IES. In this approach, an AND/OR graph is constructed to decompose and refine the sub-goals so that FRs and NFRs of a system can be identified. The FRs describe the functionality of an IES. The NFRs are the quality requirements which describe the non-behavioral aspects of an IES. After applying the goal-oriented approach, we have identified the FRs and NFRs of an IES. The list of these requirements is given below:

List of the FRs:

- FR_1: Entry of mid-semester and end-semester marks
- FR_2: Fee receipt after depositing the examination fee
- FR_3: View semester result

- FR_4: Information about the seating arrangement
- FR_5: Type of examination: Online or offline
- FR_6: Payment of online or offline examination fee
- FR_7: Upload information related to the examination
- FR_8: Generate hall ticket
- FR_9: Examination form approved by the Controller of Examinations
- FR_{10}: Fill examination form

List of the NFRs:

- NFR_1: Security
- NFR_2: Reliability
- NFR_3: Performance

4.3.3 Step 3: Evaluation of SRs by decision makers

Here, we have considered the five decision makers for evaluating the FRs and NFRs by using the following linguistic variables: "*Very weak*" (VW), "*Weak*" (W), "*Medium*" (M), "*Strong*" (S), and "*Very strong*" (VS). These variables are modeled by the TFNs for evaluation of the FRs based on NFRs. The TFN for the different linguistic variables can be identified from Figure 5. For example, the TFN for W will be (2,4,6). The NFRs are evaluated by the ten stakeholders by using the following linguistic variables: "Very high (VH)", "High (H)", "*Medium (M)*", "*Low (L)*", and "*Very low (VL)*". The TFNs for these linguistic variables can be identified from Figure 6. For example, the TFN for VH will be (0,0,0.25). The opinions of the ten stakeholders are captured after evaluating the NFRs so that the weights of the NFRs can be determined. The results after the evaluation of the NFRs by the stakeholders are exhibited in Table 1. We have adopted the data for the evaluation of the FRs by the decision makers from [28], see Table 2.

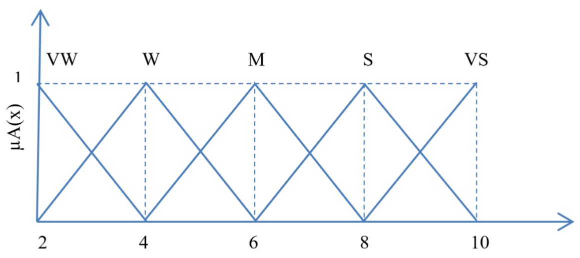


Fig.5: *Linguistic variables for the evaluation of the FRs and NFRs.*

Table 1: *Eliciting the weight of NFRs.*

NFR	Stakeholders									
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
NFR_1	VH	H	VH	H	M	L	M	VL	H	L
NFR_2	H	L	VH	VH	H	M	L	M	M	VL
NFR_3	M	VH	L	H	VH	H	VH	L	V	M

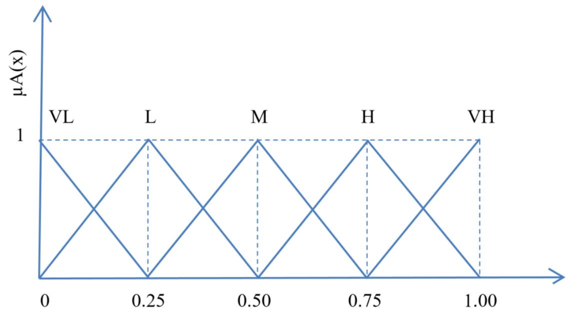


Fig.6: *Linguistic variable for eliciting the weights of NFRs.*

Table 2: *Evaluation of FRs based on NFRs.*

NFRs	FRs	DMs				
		DM_1	DM_2	DM_3	DM_4	DM_5
NFR_1	FR_1	W	S	M	S	VS
NFR_2		S	VS	S	VS	S
NFR_3		VS	VW	S	W	S
NFR_1	FR_2	S	VS	W	S	M
NFR_2		VS	S	M	VW	W
NFR_3		W	VS	S	VW	S
NFR_1	FR_3	VS	W	M	VS	VW
NFR_2		S	W	VS	M	VW
NFR_3		VW	S	VS	M	S
NFR_1	FR_4	S	VS	W	S	S
NFR_2		S	M	VS	W	M
NFR_3		M	M	S	VS	S
NFR_1	FR_5	VW	W	M	S	M
NFR_2		M	S	W	VW	S
NFR_3		S	M	W	VW	S
NFR_1	FR_6	S	VS	S	M	S
NFR_2		S	VS	S	W	VS
NFR_3		VS	W	S	M	M
NFR_1	FR_7	VW	S	M	M	M
NFR_2		M	VS	S	M	VS
NFR_3		S	S	VW	W	S
NFR_1	FR_8	S	S	W	S	S
NFR_2		VS	W	S	W	M
NFR_3		M	S	W	S	S
NFR_1	FR_9	VS	M	W	VS	S
NFR_2		S	M	VW	VS	M
NFR_3		M	S	VS	S	S
NFR_1	FR_{10}	S	S	VS	W	VW
NFR_2		S	VS	W	S	VS
NFR_3		W	S	VS	S	VS

4.3.4 Step 4: Computation of ranking values of FRs using fuzzy TOPSIS

The ranking values of the FRs are computed using the fuzzy TOPSIS method based on the data given in Tables 1 and 2. We have implemented the fuzzy TOPSIS method using the VC++ language so that the ranking order of both the small and large datasets can be identified. The graphical user interface (GUI) of the output of the program is shown in Figure 7.

Here, we explain the steps to identify the ranking values of the FRs. The $FDMat$ of ten FRs is constructed after applying equation (1) and the results are given in Table 3. The weights of the NFRs are obtained after applying equation (2). This is shown in Table 3. The $NormFDMat$ is obtained after applying equation (4). The results are summarized in Table 4. The weighted $NormFDMat$ is obtained after applying equation (5) and the results are given in

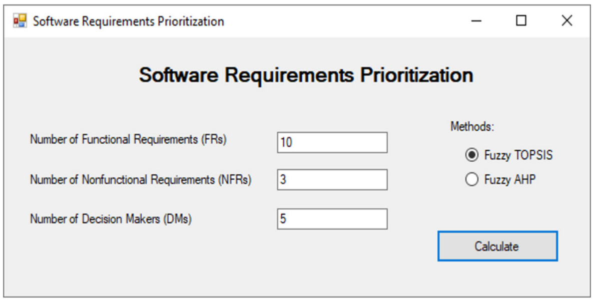


Fig. 7: *GUI for computing the ranking values of FRs.*

Table 5.

Table 3: *The FDMat of ten FRs and weights of NFRs.*

FRs	NFRs		
	NFR_1 (0,0.57,1)	NFR_2 (0,0.55,1)	NFR_2 (0,0.67,1)
FR_1	(2,6.8,10)	(6,8.8,10)	(2,6.4,10)
FR_2	(2,7.2,10)	(2,5.6,10)	(2,6.4,10)
FR_3	(2,6.4,10)	(2,6,10)	(2,6.8,10)
FR_4	(2,5.6,10)	(2,6.8,10)	(4,7.6,10)
FR_5	(2,5.2,4)	(2,5.6,10)	(2,5.6,10)
FR_6	(4,6.6,10)	(2,8,10)	(2,6.8,10)
FR_7	(2,5.6,10)	(4,8,10)	(2,6,10)
FR_8	(2,7.2,10)	(2,6.4,10)	(2,6.8,10)
FR_9	(2,7.6,10)	(2,6.4,10)	(2,7.6,10)
FR_{10}	(2,6.4,10)	(2,8,10)	(2,8,10)

Table 4: *The NormFDMat.*

FRs	NFRs		
	NFR_1	NFR_2	NFR_2
FR_1	(0.2,0.68,1)	(0.6,0.88,1)	(0.2,0.64,1)
FR_2	(0.2,0.72,1)	(0.2,0.56,1)	(0.2,0.64,1)
FR_3	(0.2,0.64,1)	(0.2,0.6,1)	(0.2,0.68,1)
FR_4	(0.2,0.56,1)	(0.2,0.68,1)	(0.4,0.76,1)
FR_5	(0.2,0.52,0.4)	(0.2,0.56,1)	(0.2,0.56,1)
FR_6	(0.4,0.66,1)	(0.2,0.8,1)	(0.2,0.68,1)
FR_7	(0.2,0.56,1)	(0.4,0.8,1)	(0.2,0.6,1)
FR_8	(0.2,0.72,1)	(0.2,0.64,1)	(0.2,0.68,1)
FR_9	(0.2,0.76,1)	(0.2,0.64,1)	(0.2,0.76,1)
FR_{10}	(0.2,0.4,1)	(0.2,0.8,1)	(0.2,0.8,1)

The distances between the FRs and $FPISol$ with respect to NFRs are computed using equation (6). The results after the computation are given in Table 6. Equation (7) is used for computing the distances between the FRs and $FNISol$ with respect to NFRs. The results are summarized in Table 7.

Equation (8) is used to compute the closeness coefficients of FRs. The results are summarized in Table 8.

Table 5: *The Weighted NormFDMat.*

FRs	NFRs		
	NFR_1	NFR_2	NFR_2
FR_1	(0,0.38,1)	(0,0.48,1)	(0,0.42,1)
FR_2	(0,0.41,1)	(0,0.30,1)	(0,0.42,1)
FR_3	(0,0.36,1)	(0,0.33,1)	(0,0.45,1)
FR_4	(0,0.31,1)	(0,0.37,1)	(0,0.50,1)
FR_5	(0,0.29,1)	(0,0.30,1)	(0,0.37,1)
FR_6	(0,0.37,1)	(0,0.44,1)	(0,0.45,1)
FR_7	(0,0.31,1)	(0,0.31,1)	(0,0.40,1)
FR_8	(0,0.41,1)	(0,0.35,1)	(0,0.45,1)
FR_9	(0,0.43,1)	(0,0.35,1)	(0,0.50,1)
FR_{10}	(0,0.42,1)	(0,0.44,1)	(0,0.53,1)

Table 6: *The distances between $FR_i(i = 1, 2, \ldots, 10)$ and FPISol with respect to each NFR.*

Distance between FRs and $FPISol$	NFR_1	NFR_2	NFR_3
$d(FR_1, FPISol)$	0.67	0.64	0.66
$d(FR_2, FPISol)$	0.66	0.7	0.66
$d(FR_3, FPISol)$	0.67	0.69	0.65
$d(FR_4, FPISol)$	0.7	0.67	0.64
$d(FR_5, FPISol)$	0.70	0.70	0.67
$d(FR_6, FPISol)$	0.67	0.65	0.65
$d(FR_7, FPISol)$	0.7	0.7	0.67
$d(FR_8, FPISol)$	0.66	0.68	0.65
$d(FR_9, FPISol)$	0.66	0.68	0.64
$d(FR_{10}, FPISol)$	0.66	0.65	0.64

Table 7: *The distances between $FR_i(i = 1, 2, \ldots, 10)$ and FNISol with respect to each NFR.*

Distance between FRs and $FPISol$	NFR_1	NFR_2	NFR_3
$d(FR_1, FNISol)$	0.61	0.64	0.62
$d(FR_2, FNISol)$	0.61	0.6	0.62
$d(FR_3, FNISol)$	0.60	0.6	0.63
$d(FR_4, FNISol)$	0.6	0.60	0.64
$d(FR_5, FNISol)$	0.6	0.6	0.60
$d(FR_6, FNISol)$	0.60	0.62	0.63
$d(FR_7, FNISol)$	0.6	0.6	0.61
$d(FR_8, FNISol)$	0.6	0.60	0.63
$d(FR_9, FNISol)$	0.62	0.60	0.64
$d(FR_{10}, FNISol)$	0.62	0.62	0.64

Table 8: *Closeness coefficients and ranking order of FRs.*

FRs	Closeness coefficients	Ranking order
FR_1	0.486	2
FR_2	0.475	8
FR_3	0.476	6
FR_4	0.477	5
FR_5‘	0.465	10
FR_6	0.484	3
FR_7	0.466	9
FR_8	0.475	7
FR_9	0.483	4
FR_{10}	0.490	1

4.3.5 Step 4: Computation of ranking values of FRs using fuzzy TOPSIS

The aim of this section is to model the FRs using UCDs and class diagrams and also discuss the formulated RQs.

RQ 1: Which set of requirements should be modeled during the SRs elicitation?

During the development process the top n requirements are selected on the basis of the ranking order for modeling and analysis. The value of n is decided by the stakeholders who are participating in the development process. If we assume that top 3 FRs will be selected for the implementation, then functional requirements FR_{10}, FR_1, and FR_6 will be modeled in the first release of software. These requirements are modeled by the UCDs in Figures 8, 9, and 10.

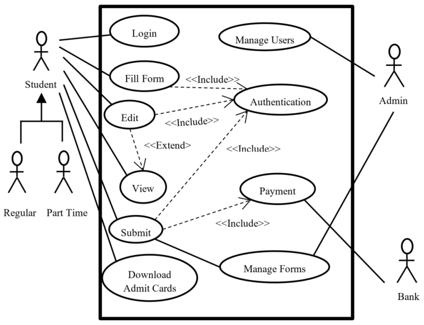


Fig.8: *Use-case diagram of* FR_{10}.

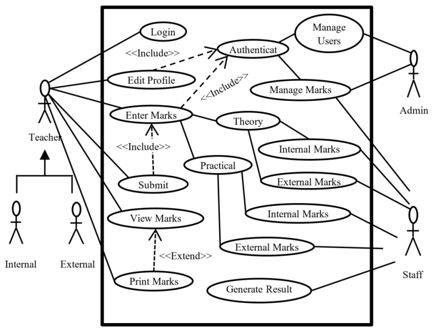


Fig.9: *Use-case diagram of* FR_1.

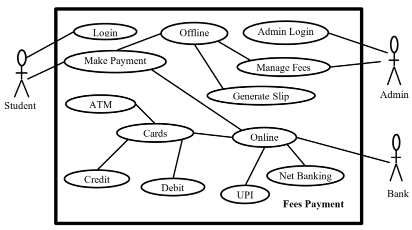


Fig.10: *Use-case diagram of* FR_6.

The same set of requirements is modeled by the class diagrams in Figures 11, 12, 13, respectively. In the case of the large dataset of an IES [27], we obtained the same ranking order of the FRs that we obtained for the small dataset, i.e., FR_{10}, FR_1, and FR_6. The ranking value of the large set of FRs was computed using a program written in VC++.

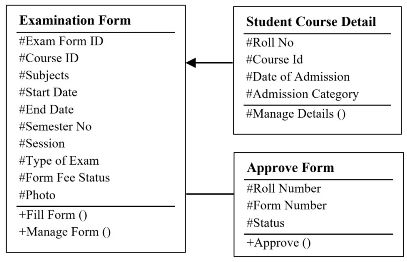


Fig.11: *Class diagram of* FR_{10}.

RQ 2: Do the UCD and class diagram capture the same information about the SRs during the requirements analysis?

The UCD represents the interaction of actors with the system. In Figure 8, three actors, student, admin, and staff, are interacting with the system for functional requirements FR_{10}. There are several different use-cases in FR_{10}: login, fill form, edit, view, submit, manage users, authentication, payments, and manage forms. In Figure 8, two actors will log into the system using "*login use-case*". These actors will be logged into the system if the use-case is successful. The use-case starts when the actors wish to enter into the examination system. If the actors enter the wrong user ID and password, the system will display an error message.

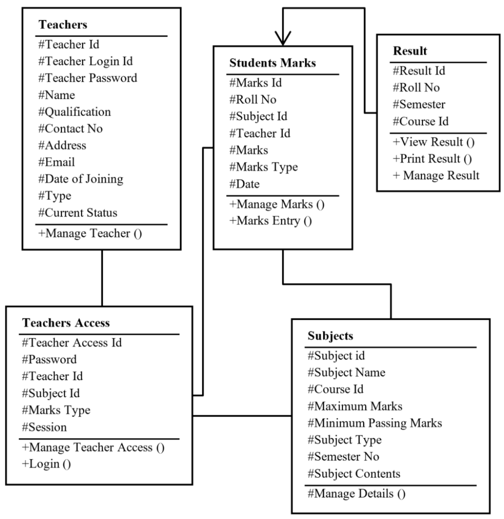


Fig.12: *Class diagram of* FR_1.

Based on the UCDs and class diagrams, we found that the requirements are more completely interpreted by the UCDs. The class diagrams are the most complex diagrams of UML and have high complexity because they include relationship and property types. Because of the complexity of the class diagrams, they may not be completely interpreted as UCDs [7, 29]. Both diagrams represent different information about

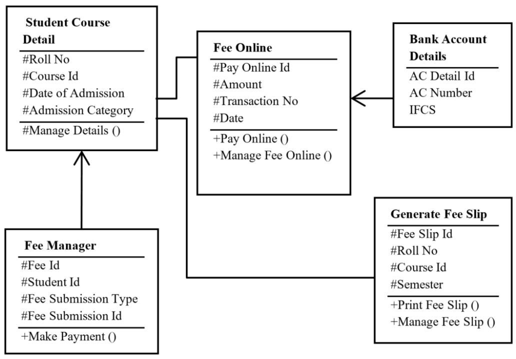


Fig.13: *Class diagram of FR_6.*

the requirements of an IES and there is no overlap in the information captured through these two diagrams. These diagrams are indispensable for the analysis of the SRs.

5. COMPARATIVE STUDY AND DISCUSSION

In Table 9, we present a comparative study between the proposed method and some selected methods based on the following criteria: (a) diagram(s) used for modeling the SRs, (b) Is there any way to select the requirements for modeling? If yes, which method is used for the selection, and (c) type of information system used for the analysis? Based on the comparative analysis we found that most of the attention in the requirements modeling domain is on applying various types of diagrams to address different issues like requirements traceability, measuring software performance using UML diagrams, generating different diagrams from use-case diagrams, etc. Both UML and goal-oriented models have been used for the modeling of SRs like use-case, class diagram, activity diagram, SysML, Tropos, etc.

Different types of information systems have been employed to model requirements like IES [5], Library Management System [30], Road Traffic Management System [31], Museum Guide mobile Information System [32], Game modeling [33] etc. In a recent study, Albaghajati and Hassine [33] introduced a game oriented use-case modeling (GOUCM) method in which UML use case diagrams have been used for modeling game related requirements. The authors implemented GOUCM to better represent the requirement of games. The applicability of the GOUCM was discussed with the following four well-known commercial games: "*Super Mario Bros*", "*Tetris*", "*Just Dance*", and "*The Walking Dead*". In their work, the main focus was only on modeling requirements of a game without discussing which set of requirements of a game will be modeled during different releases of game development. UML use case diagrams are used for analyzing the functionality of SRs. Hamza and Hammad [34] proposed an approach for generating the UML use case diagrams from text using natural language processing. The proposed method has been validated using various public software projects. Iqbal et al. [35] proposed a model-based approach using a verification and validation model. The authors have validated the proposed approach using two different case studies from the auto industry. In their work, Petri nets were used for analyzing the requirements.

Arif et al. [5] proposed a method for representing the SRs using class diagrams. The main objective of their work was to address on the following issues: (a) "*how to model the selected set of SRs using UML class diagrams*", and (b) "*is there a way to select the SRs when different stakeholders participate during decision making process under fuzzy environment?*". The proposed method was applied for modeling the FRs of an IES using use case diagrams. In [5], the authors introduced a notation to show the ranking values of class diagram for each FR. It was the first study to integrate the ranking values of the FRs in use case diagrams. The ranking values were computed using the fuzzy AHP method. One of the limitations of [5] is that only a few sets of SRs can be used for computing and modeling the SRs because fuzzy AHP has some limitations. Therefore, to extend our previous work [5], we applied the fuzzy TOPSIS method for modeling the large set of requirements of an IES using UCDs and class diagrams.

Based on our review, we found that in the literature there is no systematic approach for representing the SRs from a selected set of requirements. To address this issue, we proposed a method for representing the SRs using UCDs and class diagrams. A fuzzy based approach has been used for selecting the SRs from the list of elicited requirements. Based on our analysis, we observed that it is easier to explain the requirements to the end users with UCDs than with class diagrams. UCDs are one of the best communication tools to understand the requirements because they are easy to understand. We also observed that requirements modeling helps to elicit those FRs that we could not elicited during the requirements elicitation process. To the best of our knowledge, this is the first study which models the selected set of SRs using UCDs and class diagrams.

6. CONCLUSION AND FUTURE SCOPE

In this paper a method has been developed for modeling selected FRs of an IES using UCDs and class diagrams. In this method, fuzzy TOPSIS has been employed for computing the ranking values of the FRs based on the opinions of the decision makers. To deal with the large set of requirements, a program has been written using VC++ so that the ranking values of the requirements can be computed. After applying the goal oriented approach, we identified ten FRs and three NFRs of an IES. The opinions

Table 9: *Comparative study.*

S. N.	Methods for SRs modeling	Parameters for evaluation		
		Diagram(s) used for modeling the SRs	Is there any way to select the requirements for modeling? If yes, which method is used for the selection	Type of information system used for the analysis
1	Arif *et al.* [5]	Class diagram	Yes / Fuzzy AHP	Institute Examination System
2	Silingas and Butleris [30]	Use-case, state machine diagram, activity diagram, class diagram	No	Library Management System
3	Soares *et al.* [31]	SysML diagrams	No	Road Traffic Management System
4	Ali *et al.* [32]	Tropos goal model	No	Museum Guide mobile Information System
5	Albaghajati and Hassine [33]	Use-case driven	No	Game Modeling
6	Proposed method	Class and Use-case diagrams	Yes / Fuzzy TOPSIS	Institute Examination System

of the FRs in the context of NFRs are captured by five decision makers. These were used as an input for computing the ranking order of the FRs. Based on our analysis, we found that UML class diagrams, UCDs, activity diagrams, SysML, Tropos, etc. are widely used for representing the SRs in the literature of software engineering. Clearly UCDs are suitable tool for expressing the requirements of a system. Both UCDs and class diagrams are valuable for the analysis of the requirements. In the future, we shall focus on the following:

1. Analysis of FRs and NFRs of IES using a goal oriented approach and UML diagrams.
2. Comparing structural and behavioral UML diagrams for large set of the requirements of an IES.

References

[1] W. B. A. Karaa, Z. B. Azzouz, A. Singh, N. Dey, A. S., Ashour, and H. B. Ghazala, "Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements," *Software Practice and Experience*, Vol. 46, No. 11, pp. 1443-1458, 2016.

[2] M. Sadiq and Jain S. K., "An Insight into Requirements Engineering Processes," *3rd International Conference on Advances in Communication, Network, and Computing LNCSIT-Springer*, pp. 313-318, Chennai, February 24-25, 2012, India.

[3] M. Arif, C. W. Mohammad and M. Sadiq, "Software requirements modeling: a systematic literature review," in *IEEE International Conference on Computing, Power and Communication Technologies*, Greater Noida, India, pp. 194-200, 2020.

[4] G. Booch., J. Rumbaugh, I. Jacoboson, "The unified modeling language user guide," *Second Edition*, pp. 1-496, Addison Wesley, 2005.

[5] M. Arif, C. W. Mohammad and M. Sadiq, "Modeling of selected set of software requirements using UML class diagrams," *17th IEEE India Council International Conference*, pp. 1-6, 2020.

[6] K. Siau, L. Lee, "Role of use case diagram in requirement analysis," in *Proceedings of Americas Conference on Information Systems 2001*, Boston, USA 2001.

[7] K. Siau, L. Lee, "Are use case and class diagrams complementary in requirements analysis? An experimental study on use case and class diagrams in UML," *Requirements Engineering*, Vol. 9, pp 229– 237 2004.

[8] N. Uraipan, P. Praneetpolgrang, and T Manisri, "Application of a fuzzy Analytic Hierarchy Process to Select the Level of a Cyber Resilient Capability Maturity Model in Digital Supply Chain Systems," *ECTI Transactions on Computer and Information Technology*, vol.15, no.2, pp. 198-207, August 2021.

[9] M. M. Salih, B.B. Zaidan, A.A. Zaidan, and M. A. Ahmed, "Survey on fuzzy TOPSIS state-of-the-art between 2007 and 2017," *Computers and Operation Research*, vol. 104, pp. 207-227, 2019.

[10] J. Chanda, A. Kanjilal, S. Sengupta and S. Bhattacharya, "Traceability of requirements and consistency verification of UML use case, activity and Class diagram: A Formal approach," *2009 Proceeding of International Conference on Methods and Models in Computer Science*, pp. 1-4, 2009.

[11] K. Lano, S. K. Rahimi, "Case study: Class diagram restructuring," *Proceeding of Sixth Transformation Tool Contest*, Vol. 135 of EPTCS, Open Publishing Association, pp. 8–15, 2013.

[12] C. R. Narawita, K. Vidanage, "UML Generator – Use Case and Class Diagram Generation from Text Requirements," *International Journal on Advances in ICT for Emerging Regions*, Vol.10, No.1, pp.1–10, 2018.

[13] A. Mohammed, F. Bouchaib, M. Bassim, B. Nacir, "An effective approach for transforming use cases specifications into class diagram," *Advanced Science Letters, American Scientific Publishers*, Vol. 22, No. 10, pp. 2972-2976, 2016.

[14] A. Alsaadi, "A performance analysis approach based on the UML class diagram," *ACM SIGSOFT Software Engineering Notes*, Vol. 29, Issue 1, pp 254–260, 2014.

[15] M. Singh, A. K. Sharma, R. Saxena, "Formal

Transformation of UML Diagram: Use Case, Class, Sequence Diagram with Z Notation for Representing the Static and Dynamic Perspectives of System," *Proceedings of International Conference on ICT for Sustainable Development. Advances in Intelligent Systems and Computing*, Vol. 409, Springer, Singapore, 2016.

[16] R. Fauzan, D. Siahaan, S. Rochimah and E. Triandini, "Use Case Diagram Similarity Measurement: A New Approach," *12th International Conference on Information & Communication Technology and System (ICTS)*, pp. 3-7, 2019.

[17] F. Javed, M. Mernik, B.R. Bryant, J. Gray, "A grammar-based approach to class diagram validation," *4th International Workshop on Scenarios and State Machines: Models, Algorithms and Tools*, (2005)

[18] D. Faitelson and S. Tyszberowicz, "UML Diagram Refinement (Focusing on Class-and Use Case Diagrams)," *2017 IEEE/ACM 39th International Conference on Software Engineering*, pp. 735-745, 2017.

[19] J. S. Thakur and A. Gupta, "Automatic generation of sequence diagram from use case specification," in *Proceedings of the 7th India Software Engineering Conference, Association for Computing Machinery*, New York, pp. 1–6, 2014.

[20] M. Sadiq, A. Parveen, and S. K. Jain, "Software requirements selection with incomplete linguistic preference relations," *Business and Information Systems Engineering*, Springer, pp. 1-20, 2021

[21] M. Sadiq, "A fuzzy-set based approach for the prioritization of stakeholders on the basis of the importance of software requirements," *IETE Journal of Research, Taylor and Francis*, Vol. 63, Issue 5, pp. 616-629, 2017.

[22] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, 8, pp. 338–353, 1965.

[23] L. A. Zadeh, "The concept of a linguistics variable and its application to approximate reasoning (I)," *Inf. Sci.*, 8, pp. 199–249, 1975.

[24] S. Kamlu, and V. Laxmi, "Implementation of fuzzy model for maintenance scheduling of vehicles based on Monte Carlo simulation and geographical information system," *IETE. Journal of Research*, Vol. 63, no. 2, pp. 225-237, 2017.

[25] Horkoff J., Aydemir F. B., Cardoso E., Li T., Mate A., Paja E., Salnitri M., Mylopolous J., Giorgini G., "Goal oriented requirements engineering-a systematic literature map," *24th International Requirements Engineering Conference*, pp. 106-115, 2016.

[26] P. Runeson, M. Host, A. W. Rainer, and B. Regnell, "Case study research in software engineering," in *Guidelines and examples.* Hoboken, NJ, Wiley, 2012, pp. 1–237.

[27] M. Sadiq and V. S. Devi, "Prioritization and selection of the software requirements using rough-set theory," *IETE Journal of Research*, pp. 1-18, 2021.

[28] M. Sadiq, and S. K. Jain, "Applying fuzzy preference relation for requirements prioritization in goal-oriented requirements elicitation process," *Int. J. Syst. Assur. Eng. Manag.*, Vol. 5, no. 4, pp. 711–23, 2014

[29] K. Siau and Q. Cao, "Unified modeling language-a complexity analysis," *Journal of Database Management*, Vol. 12, Issue 1, pp. 26-34, 2001.

[30] D. Silingas and Butleris R, "UML-intensive framework for modeling software requirements," *International Conference on Information Technologies*, Kaunas, 2008, pp. 334-342.

[31] M. S. Soares, J. Vrancken, A. Verbraeck, "User requirements modeling and analysis of software-intensive systems," *The Journal of Systems and Software*, Vol. 84, Issue 2, pp. 328-339, 2011.

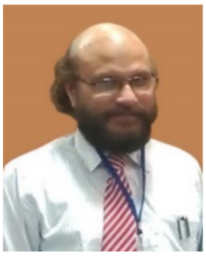
[32] R. Ali, F. Dalpiaz, P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis," *Requirements Engineering*, Vol. 15, pp. 439-458, 2010.

[33] A. Albaghajati and J. Hassine, "A Use case driven approach to game modeling," *Requirements Engineering*, vol. 27, pp. 83-116, 2022.

[34] Z. A. Hamza and M. Hammad, "Generating UML use case models from software requirements using natural language processing," *International Conference on Modeling Simulation and Applied Optimization*, pp. 1-6, 2019.

[35] D. Iqbal, A. Abbas, M. Ali, M. U. S. Khan and R. Nawaz, "Requirement validation for embedded systems in automotive industry through modeling," *IEEE Access*, vol. 8, pp. 8697-8719, 2020.

Mohd. Arif joined Ph.D. in Computer Science and Technology in 2017 at the Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (JMI), A Central University, New Delhi, India. He received M. Tech. degree in Computer Science and Engineering from Maharshi Dayanand University Rohtak in 2015. He did Bachelor of Engineering in Computer Engineering and Diploma in Computer Engineering from JMI, New Delhi, India, in 2009 and 2005, respectively. He was intern at Bureau of Indian Standards, Delhi, from January 2006-January 2007. He worked around five years as a Software Developer in various Software Industries like Percentile Solutions, Web Planet, DZ Webmicro Systems Pvt. Ltd., etc; and taught different subjects in the academic session 2015-2017 as a Guest Teacher of Computer Engineering at JMI, New Delhi, India. His research interest includes Software Requirements Modeling and Architecture. He has presented his Ph. D research work in the following Conferences: IEEE GUCON-2020, India and IEEE Indicon-2020, India.



Chaudhary Wali Mohammad is a Professor of Applied Mathematics and Computing in the Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (JMI), A Central University), New Delhi, India. Prof. Wali received the B. Sc. (Hons.), M. Sc., M. Phil., and Ph. D., all degrees in Mathematics, from the Department of Mathematics, Faculty of Natural Science, Aligarh Muslim University, Aligarh, U.P., India, in the year, 1978, 1980, 1983, and 1986, respectively. After completion of the Ph.D. degree in Mathematics, Prof. Wali joined the JMI, New Delhi, on October 1, 1986, as a Lecturer (Contract) in Mathematics and become permanent faculty as a Lecturer (Mathematics) on February 9, 1989, in the Faculty of Engineering and Technology, JMI, New Delhi. Prof. Wali promoted as a Reader (Mathematics) in 1999 and become full Professor on February 9, 2007. He worked as Head of the Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, JMI, New Delhi since December 5, 2014 to December 4, 2017. His area of interest includes Multiple Hypergeometric Functions, Data Structure and Algorithms, Graph Theory, Multi-Criteria Decision-Making algorithms and its application to Software Engineering. He has supervised three Ph.D. Thesis in the area of Multiple Hypergeometric Functions and three Ph.D. Thesis in the area of Computer Science and Technology. He has published his research work in the following reputed journals: *Alexandria Engineering Journal, Elsevier, 2022; International Journal of Information Technology, 2018, Springer; International Journal of Computers and Applications*, 2020, Taylor and Francis; *Global Journal of Science Frontier Research*, 2010, USA, etc.

Mohd. Sadiq is working as an Assistant Professor and Section Incharge-Head of Computer Engineering Section, UPFET, Jamia Millia Islamia, A Central University, New Delhi, India. He was a Post-Doctoral Fellow at the Pattern Analysis and Machine Intelligence Laboratory, Department of Computer Science and Automation, Indian Institute of Science (IISc) Bangalore, India, from July 2019 to June 2021. During his research work, he taught Soft Computing and Intelligent Agents jointly with Prof. V. Susheela Devi at IISc Bangalore. In 2017, he received a Ph.D. degree in Computer Engineering from the National Institute of Technology, Kurukshetra, India. Prior to joining IISc Bangalore, he has worked as a Post-Doctoral fellow at the Center for Soft Computing Research, Indian Statistical Institute (ISI), Kolkata, India; and also, as a Visiting Scientist at the Systems Science and Informatics Unit, ISI, Bangalore Centre, India. Dr. Sadiq has delivered expert lectures on Artificial Intelligence and Compiler Design as a visiting faculty at the Indian Institute of Information Technology Lucknow (IIIT), India. He has published his research work in the following reputed journals: *Expert Systems with Applications*, Elsevier, 2022; *Alexandria Engineering Journal*, Elsevier, 2022; *Business & Information Systems Engineering*, 2021, Taylor and Francis; *IETE Journal of Research*, 2017 and 2021, Taylor and Francis; *International Journal of Information Technology*, 2021, Springer; *International Journal of System Assurance Engineering and Management*, 2014, 2015, and 2021, Springer; *International Journal of Computers and Applications*, 2020, Taylor and Francis; *International Journal of Business Information Systems*, 2020, Inderscience, and *CSI Transactions on ICT*, 2013, Springer. His research interests lie at the intersection of software engineering and fuzzy /rough set-based multicriteria decision-making algorithms.