Type -1 Fuzzy logic based system for predicating the best Combination of Requirements Elicitation Techniques

Khalid Almohammadi

Computer Science department, Community College, University of Tabuk, Tabuk, Saudi Arabia

Summary
Requirements Engineering (RE) is comprised of many phases. The first phase is the Requirements elicitation, which constitutes the most important phase. There have been a lot of advancements in methods that can support the elicitation phase. Each of the methods, however, has own advantages and disadvantages. These characteristics makes these methods largely varied and hence the difficulty in the selection of an appropriate method. Analysts often select methods with which they have experience or of their preference and in most cases do not consider the characteristics of the project and stakeholders. This paper proposes a type-1 fuzzy logic system which can learn the best combination of requirements elicitation techniques based on factors affecting this process and related to the characteristics of the project and stakeholders to aid novice analysts later on in the selection process. To this end, we carried out experiments on 30 projects and examined the input of 4 expert analysts who served as study participants. Our proposed system could predict a combination of methods for requirements elicitation at lower average errors. This means that the system can handle the potential uncertainties.

Key words:
type-1 fuzzy logic, requirements engineering, requirements elicitation

1. Introduction

Providing adequate solutions to customers' issues entails the need for all stakeholders to agree [1], [2]. As such, the proposed system has a requirement to mean what needs to be made true. A "requirement" represents how ideas are to be incorporated in a system or an application being developed [3]. It is an expression of system service or a description of user-level characteristics, the system at large, individual limitations, and the clients' needs [3]. However, the term may also be used to describe the qualities of a system as well as how it behaves [4]. The success of a project or system being developed depends tremendously on the quality and accuracy of the requirements [5], [6]. To facilitate the understanding of everyone involved, natural language is used to express the requirements phase [3].

Analysts frequently depend on the requirements to understand critical elements and functions while developing a given project. Requirements contribute significantly to the other phases of software product development, including the design, execution, and validation. As such, an inappropriate requirements elicitation and poor management processes regarding requirements can easily contribute to a failure in software development [7].

Most software solutions fail due to their inability to meet the needs of the stakeholders [8]. Such failures are enormously expensive and convey a tremendous risk, which could exponentially be a threat to lives. Because its failure would pose the likelihood of failure, inherent challenges, and other risks to a project, requirements engineering needs to be well considered as an aspect of software development. Such failures are accompanied by tremendous costs [3]. Software project failures and overruns are an additional limitation in highly competitive and modern environments. Such scenarios taint company images adversely. Companies may also lose goodwill and experience reduced revenue drives because of a decrease in intended customer and client satisfaction [1–3]. According to reports by the Standish Group, in 2009, only 34% of the software projects were successful: 44% of the projects experienced challenges and 22% failed [9]. The Chaos Report of 1995 indicated that requirements engineering activities positively influenced the success of more than 42% of the successful projects [10]. Poor requirements engineering activities were attributed as the main reason for the failure of 43% of the software failures. Previous researchers also observed a 70% difficulty in identification of the requirements and 54% in unclear and poorly organized requirements [11–12].

In an attempt to mitigate this problem, methods were advanced to perform a range of activities. These methods included eliciting the requirements of the stakeholders, analyzing the impacts of these requirements on the project or software to be developed, pointing out the specific software product for the stakeholder's needs, and counter-checking the acceptability of the project. Interviews, questionnaires, observations, and prototyping constitute a range of elicitation methods [13], [14], [15]. All of these methods have individual weaknesses and strengths, and

Manuscript received January 5, 2020
Manuscript revised January 20, 2020

none of them are suitable for all situations. Although they work in different situations, in some cases, they can work in a complementary association [16]. As such, the disadvantages of one method can be neutralized by the advantages of a second one [17]. The use of multiple methods assures the possibility of knowing as many requirements as possible, hence contributing to reliable requirements elicitation [17]. Nonetheless, it is challenging when selecting a method or a combination of methods for a given project. If the selected method is inappropriate, the requirements elicitation process will not be effective. When selecting a requirements eliciting method, in most cases, personal preferences are used instead of selecting a method based on the features of the project, the method, or the stakeholder needs.

There are some common reasons that guide the selection of a given method by software engineers [18]: The method is the only one they know; the method is their most preferred in all scenarios; also, some engineers follow a procedure that specifies a given method, and some will guess on the effectiveness of a method based on the prevailing situation. Such criteria of deciding on a method can reduce the accuracy of the requirements elicitation process significantly, leading to the elicitation of poor quality requirements, which would not suit the requirements of the stakeholders. Therefore, an efficient requirements elicitation procedure is imperative when selecting the elicitation method by developing a data-driven model that limits the human subjectiveness and aids in the selection of the most appropriate methods for elicitation. This would include learning the best combination of elicitation methods according to the factors such as the size of the stakeholder, the users' computer skills, the stakeholders' diversity, the presence of requirements that can be reused, the availability of stakeholders, and the time limitations for the development [19], [20].

In the paper, we advance a novel type-1 fuzzy logic system for the selection of the best combination of elicitation methods. The system is based on several factors that influence the elicitation process and are dependent on the features of both the software and the stakeholder. These factors include the time limitations for the development of the software, the available reusable requirements, and the availability and size of stakeholders. The proposed system can gather these inputs and outputs and learns the appropriate combination of requirements elicitation methods and then make the prediction based on the project and the features of the stakeholders. We conducted experiments on 30 projects and collected the analyses of 4 expert analysts. Our system predicted combinations of methods for requirements elicitation with lower average of errors. The system was hence able to counter the uncertainties that were encountered in the selection process.

This paper is divided into six sections. The remaining are as follows: Section 2 provides an overview of the machine learning techniques used for requirements elicitation. Section 3 describes type-1 fuzzy logic systems. Section 4 describes the proposed theoretical and practical type-1 fuzzy logic system for predicating the best combination of requirements elicitation techniques. Section 5 highlights the experiments and the results. Finally, Section 6 outlines the conclusion and future work.

2. An overview of the machine learning techniques used for requirements elicitation

Requirements elicitation is a critical step in the development of any new software application [14]. Most software systems fail simply because of poor elicitation activities. A requirement can be used to mean the set of requests of a need to be fulfilled [14]. In software engineering, it is used to describe what software systems need to do. Since a typical system's requirements may number in the thousands, it is impossible to deduce the requirements of a system without a proper elicitation process. There is some degree of bias in the process of selecting a requirements elicitation method, and there is no evidence based on an optimized combination of requirements elicitation methods. Consequently, there is the need to develop a data-driven model that limits the human subjectiveness and aids in the selection of the most appropriate methods for elicitation. This selection would be based on method features, stakeholders, and software features. Since analysts lack adequate information regarding elicitation methods, they frequently experience difficulties when selecting an elicitation method for a given software project. Machine learning techniques model similar behavior through learning the skills and expertise of the analysts while relying on the features of the elicitation method, the stakeholders, and the software with uncertain data.

Theoretical approaches have been widely proposed to direct analysts while selecting an elicitation method [1], [2], [14], [19]. By learning the selection decisions made by expert analysts, limited artificial intelligence methods have gained implementation in requirements elicitation. A fuzzy logic model was advanced to choose the appropriate requirements elicitation methods depending on their characteristics per the stakeholders' and the project situations. The function of [21], the proposed grading methods according to the fuzzy logic and average, is to simplify the subjective choice of an elicitation method selection. A section model based on an artificial neural network was proposed for a suitable elicitation method for a project in [16]. Human involvement was hence limited in the selection. The model was executed through a neural network fitting tool in MATLAB and recorded 81% accuracy.

3. Type-1 Fuzzy logic systems methodology

Fuzzy systems have been widely applied as information processing technologies in industrial and technical fields, including such as control, automation, image signaling, pattern recognition, and knowledge mining [22], [23]. Traditional mathematical tools are limited in their application in situations when knowledge regarding a condition is vague, uncertain, and not imprecise [22]. Therefore, FLSs offers a platform for the design of massive regulators that can produce better performance when handling uncertainties and impression characteristics in real-world environments. The information on FLSs platforms is displayed in modes that can be easily read, thereby easing their integration. Such FLSs are type-1 and they use defined type-1 fuzzy sets.
Zadah first demonstrated fuzzy logic in 1965 [24]; since that time, it has gained in popularity and has been considered the most appropriate user-modeling tool because of its ability to replicate human thinking [25]. Since the expressions can be partially true, between truth and falsity, fuzzy logic is a continuation of ancient theory [23]. Fuzzy logic systems are comprised of four phases (see Figure 1): fuzzifier, rule base, inference engine, and defuzzifier [23]. Numerical data can be fed into the system as a source of the rules, or in other cases, the rules can be obtained from experts. Once the rules are established in the system, the FLS will be considered to be mapping, from defined input to defined outputs. The mapping process can be represented numerically as y = f (x) [23]. The FLS can be used to include a consideration for the factors that influence the choice of the appropriate elicitation method. FLS modeling systems that are based on expert analysts simplify the reasoning for software designers and customers. These are generally easy to use and their significance can be appreciated.

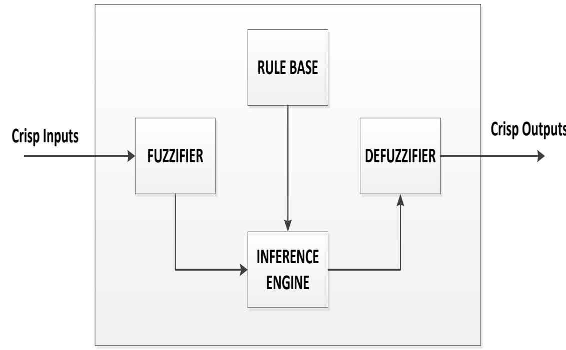

Fig. 1 The structure of a fuzzy logic system (adopted from [26]).

4. The Type-1 Fuzzy Logic System for Predicating the best Combination of requirements Elicitation Techniques

The proposed system will have the ability to learn various suitable combinations for the elicitation requirements techniques where the system starts to gather the outputs and inputs during the first phase. Therefore, the proposed system will learn and create a descriptive and suitable model following the project's requirements by the stakeholders and their associated best elicitation requirements techniques, selected by the expert analysts through the generation of desired output and input pairs of data, as shown below [27], [28], [29].

$$x^{(t)}; y^{(t)} \qquad (t = 1.2 \ldots N) \tag{1}$$

Figure 2 illustrates the proposed system in which the gathered data and N in the first equation is recognized as a series of data examples, $x^{(t)} \in R^n$, and $y^{(t)} \in R^k$. The system has a combination of rules explaining the manner in which the k output, which is a requirements elicitation technique, variables $y = (y_1, \ldots, y_k)^T$ are influenced by the variables from the inputs representation on the project (software) and the characteristics of the stakeholders $x = (x_1, \ldots, x_n)^T$. The input and output mapping model is achieved using established fuzzy rules that do not require any given mathematical model, hence single rules are changed online, which affects particular aspects used in the descriptive model being learned and created in the system [27], [28], [29].

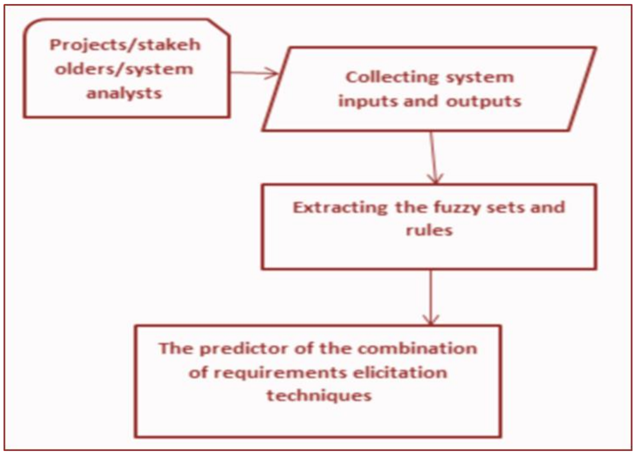

Fig. 2 The structure of a fuzzy logic system (adopted from [26]).

The extracted data is analyzed and examined using the functions available on the fuzzy logic sets relating to the inputs and the outputs. The system also uses an unsupervised one-pass technique (inspired from [27], [28], [29]) in the extraction of rules using the collected data, which additionally assists in the description of the best requirements elicitation techniques that analysts can select in the building of a model with the capacity of learning the different factors, both for the stakeholders and the projects. In addition, the learned inputs are to be considered and will be used to create an output that considers the current inputs. The FLS will also have the capacity to modify the learned rules to enable online enhancement and improvement of the different rules (see Figure 2).

5. Experiments and Results

This study was conducted using a sample of 4 analyst experts and 30 projects: The given tests included evaluations of the requirements elicitation techniques suitable for multiple stakeholders and the software characteristics. The experts reviewed the following parameters while collecting assessments on the inputs: Time constraint to come up with the project, the size and availability of stakeholders, the familiarity of computing by the users, and the available reusable requirements. The variables were collected, and the results were presented to the analysts to help determine the different combinations of linguistic variables (as a percentage) for the best four elicitation techniques, including interviews, prototyping, user observation, and questionnaires.
The four outputs were recorded by the system. After the data from the output and input were collected in the phase, the type-1 fuzzy models were built and fuzzy sets were obtained to capture the uncertainty signifying each expert analyst's perspective relating to a particular linguistic label and how it explains the elicitation requirements technique and the related impacted inputs [30]. To accomplish this, the experts were asked to model and represent different linguist variables from their own point-of-view and then analyze them further using [30]. Figure 4 shows where it illustrates the extracted type-1 fuzzy set, and the rules are generated from all of the expert analysts using [27], [28], [29], [30], as explained in Section 3. An example from the extracted rules is shown in the following figure.
If the size of the stakeholders is large and the users' computer skills are high and the presence of requirements that can be reused is true and the availability of the stakeholders is limited, and the time limitations for the development is short, then the recommendation for using interviews is low, for using observations is very low, and for using questionnaires is very high, and for using prototyping is high.

If the size of the stakeholders is large and the users' computer skills are high and the presence of requirements that can be reused is true and the availability of the stakeholders is limited, and the time limitations for the development is short, then the recommendation for using interviews is low, for using observations is very low, and for using questionnaires is very high, and for using prototyping is high.

Fig. 3 An example from the extracted rules

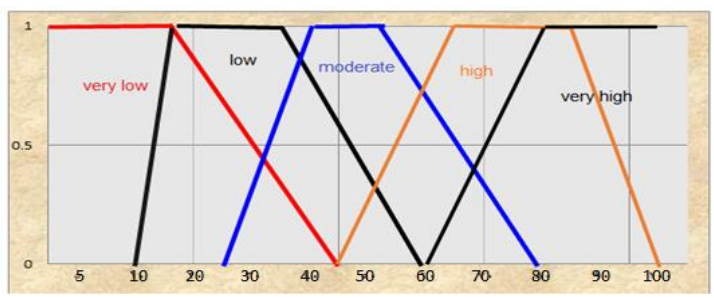

Fig. 3 Type-1 fuzzy sets for the outputs from the expert analysts' perspective

The testing results of the predicated four outputs after training the proposed model illustrate that the average errors of the type-1 fuzzy logic system are 0.37, 0.39, 0.31, and 0.44, respectively for the four outputs "interview, observations, questionnaires and prototyping, indicating that the type-1 system appropriately models the experts' selection and the best combination of requirements elicitation expertise behavior. The system resulted in better requirements elicitation techniques behavior modeling after using the type-1 fuzzy logic system.

6. Conclusions and Future work

Requirements elicitation is a critical phase in requirements engineering. Several techniques have been proposed that support the elicitation technique. Since different processes exhibit various strengths and weaknesses, methods with a combination of output and input yield the best results. Analysts select techniques based on preferences and not necessarily the attributes of the project, the stakeholders, or the methods. This paper proposed a novel data driven type-1 fuzzy logic system for the best combination elicitation techniques selection, which was made considering factors affecting the processes of elicitation requirements. The proposed system is flexible when it comes to capturing the outputs and inputs in the predictions of the appropriate elicitation requirements techniques in which the system learns the suitable combination of elicitation technique using the features of the projects and stakeholders. Real-world experiments were performed using 30 projects and the evaluations of four analyst experts were examined. The system was able to appropriately predict and forecast a combination of elicitation requirements techniques using the type 1 fuzzy logic system, hence illustrating that the method could capture encountered uncertainty.

In future work, this system will be expanded; a type-2 fuzzy logic system will be applied with more inputs and outputs and then compared with this type-1 proposed system. In addition, more real world evaluations will be used in measuring the success of software and their associated time using our systems.

References

[1] Hussain, A., Mkpojiogu, E.O.C., Kamal, F.M. (2015), Eliciting user satisfying requirements for an e-health awareness system using kano model. Proceedings of the 14th WSEAS International Conference on Computer and Computational Science (ACACOS'15), Kuala Lumpur, Malaysia

[2] Hussain, A., Mkpojiogu, E.O.C., Abdullah, I. (2016a), Investigation of Current Requirements Engineering Practices Among Software Developers at the Universiti Utara Malaysia Information Technology (UUMIT) Centre. 1st International Soft Science Conference) ISSC'16), Langkawi Island, Malaysia, 11-13 April, 2016..

[3] A. Hussain, E.O.C. Mkpojiogu and F.M. Kamal, The role of requirements in the success or failure of software projects.International Review of Management and Marketing, 6(S7),306-311(2016); ISSN: 2146-4405

[4] Inam, A. (2015), A Study of Requirements Engineering Practices Among Software Developers at UUM Information Technology. MSc. Dissertation Report. Malaysia: Universiti Utara Malaysia

[5] Gupta, S., Wadhwa, M. (2013), Requirement engineering: An overview. International Journal of Research in Engineering and Technology, 1(2), 155-160

[6] Krauss, E. (2012), Requirement Engineering and Project Management. Heidelberg: Springer

[7] Pfleeger, S.L., Atlee, J.M. (2006), Software Engineering: Theory and Practice. London, UK: Pearson, India

[8] S. Fricker, R. Grau, and A. Zwingli, "Requirements Engineering: Best Practice," in Requirements Engineering for Digital Health, S. Fricker, C. Thuemmler, and A. Gavras, Eds., ed: Springer, 2014

[9] S.K. Kamuni, Study of Factors that Induce Software Project Overrun Time, Mechanical and Manufacturing Engineering, Paper 10, (2015).

[10] A. Hussain, E.O.C. Mkpojiogu and Husin, Z. Requirements: Towards an understanding on why software projects fail. 1st International Soft Science Conference, 11-13 April 2016, Langkawi, Malaysia, (2016).

[11] S. Asghar and M. Umar, Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components. Int'l J. Soft. Eng., 1(1), 32-50 (2010)

[12] D.C. Gause and G.M. Weinberg, Exploring requirements: quality before design. (Dorset House, NY, 1989).

[13] Ann M. Hickey, Alan M. Davis, "Elicitation Technique Selection: How Do Experts Do It", Colorado: IEEE, 2003

[14] Shadab Khan[1], Aruna B Dulloo[2] and Meghna Verma[3], "Systematic Review of Requirement Elicitation Techniques", India, 2014

[15] O. Mrayat, N. Norwaw and N. Basir, "Requirements Elicitation Techniques: Comparative Study," International Journal of Recent Development in Engineering and Technology, vol. 1, issue. 3, 2013.

[16] N. R. Darwish, A. A. Mohamed and A. S. Abdelghany "A Hybrid Machine Learning Model for Selecting Suitable Requirements Elicitation Techniques", IJCSIS, Vol.14, No.6, 2016, pp. 380-391.

[17] M. Yousuf and M. Asger, "Comparison of Various Requirements Elicitation Techniques," International Journal of Computer Applications, vol. 116, vol. 4, 2015.

[18] D. Carrizo, O. Dieste and N. Juristo, "Systematizing requirements elicitation technique selection," Information and Software Technology, vol. 56, pp. 644-669, 2014.

[19] Ayalew, Yirsaw & Abebe, Semahegn. (2005). A Fuzzy Logic-based Approach for Requirements Elicitation Techniques Selection.. Proceedings of the 2005 International Conference on Software Engineering Research and Practice, SERP'05. 1. 448-456.

[20] L. Macaulay, C. Fowler, M. Kirby, A. Hutt, "USTM: An Approach to Requirements Specification," Interacting with Computers, pp. 92-118, 1990.

[21] Wajid Abbasi , 'Fuzzy Logic based intelligent software Requirement Elicitation Techniques Selection model', MSSE thesis, Iqra University, Islamabad,2013 .

[22] J. M. Mendel, “Uncertain Rule-Based Fuzzy Logic Systems,” Prentice Hall 2001.
[23] J. M. Mendel, “Fuzzy logic systems for engineering: a tutorial,” Proceedings of the IEEE, vol. 83, no. 3, pp. 345–377, March 1995.
[24] L. A. Zadeh, “Fuzzy sets,” Inf. Control, vol. 8, pp. 338–353, 1965.
[25] J. Bih, “Paradigm shift -an introduction to fuzzy logic,” IEEE Potentials, vol. 25, no. 1, pp. 6–21, 2006.
[26] K. Almohammadi and H. Hagras, “An Interval Type-2 Fuzzy Logic Based System for Customised Knowledge Delivery within Pervasive E-Learning Platforms,” Proceeings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2872–2879, October 2013.
[27] L. X. Wang, “The MW method completed: A flexible system approach to data mining,” IEEE Trans. Fuzzy Syst., vol. 11, no. 6, pp. 768–782,Dec. 2003.
[28] L. X. Wang and J. M. Mendel, “ Generating fuzzy rules by learning from examples,” IEEE Trans. Syst. Man Cybern., vol. 22, no. 6, pp. 1414–1427,Nov./Dec.1992.
[29] F. Doctor, H. Hagras, and V. Callaghan, “A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments,”IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 35, no. 1, pp. 55–65, Jan. 2005.
[30] K. Almohammadi and H. Hagras, “An adaptive fuzzy logic based system for improved knowledge delivery within intelligent E-Learning platforms,” Proccedings of the the 2013 IEEE International Conference on Fuzzy Systems, 2013, pp. 1–8.

Khalid Almohammadi completed his bachelor’s degree in computer science from Taibah University in the Kingdom of Saudi Arabia. He then pursued a profession in teaching at Tabuk University. Building on this experience, he was awarded a scholarship to complete his master’s in computer science (MSc) at Newcastle University in England, which he obtained with distinction in 2011. He recently completed his PhD in computer science from Essex University in England, which enables him to incorporate all his previous knowledge and experience in education and e-learning. The focus of his research is the development of theoretical and practical environments to enhance students’ learning experience and engagement by using fuzzy logic systems. He currently works as an assistant professor in the Computer Science Department at Community college Tabuk University.