A Proposal of Requirements Specification Process for Adaptive Systems Based on Fuzzy Logic and NFR-Framework

João DionisioParaiba
FACEN – Faculty of Exact and Natural Science
Methodist University of Piracicaba (UNIMEP)
Piracicaba, SP – Brazil
jdparaiba@gmail.com

Luiz Eduardo G. Martins
UNIFESP – Science and Technology Department
São José dos Campos, SP – Brazil
legmartins@unifesp.br

***Abstract*— Fuzzy Logic is a concept that deals with ambiguities, uncertainties and vague information on the solution of problems. NFR-Framework deals with the non-functional requirements which also are, very often, vaguely and full of uncertainties. In this paper, we use these concepts to propose a process for requirements specification of adaptive systems, called PERSA - Portuguese acronym to "Processo de Especificação de Requisitos para Sistemas Adaptativos". Adaptive systems consist of functional and non-functional requirements, which hold the capacity to modify themselves during the runtime with little or no human intervention at all. However, despite being a very discussed topic in Requirements Engineering (RE) community, it still lacks tools and techniques to standardize its modeling. The proposed process is instantiated in a case study which is discussed along this paper.**

Keywords-Adaptive Systems, Adaptive Requirements, Requirements Specification, Fuzzy Logic, NFR-Framework.

I. INTRODUCTION

The continuous evolution of software systems, the increase in complexity and the integration of technology, among other factors, lead the Requirements Engineering (RE) community to seek inspiration in some related areas (Robotics, Control Theory and Biology), in the attempt of finding innovative approach to the building and management of software systems. Therefore, adaptive systems are able to set their behavior at runtime as an answer to the environment and to the system itself, making it a very discussed theme in the RE community [1]. Adaptive systems have grown in importance with the increasing complexity of software systems and the need of such systems to be versatile, flexible, reliable, robust, recoverable, customizable, self-sustained and optimized, since they deal with these characteristics and with uncertain contexts which are often not discussed in the specification process, then requiring the system to adapt to unexpected changes. Adaptive system is a new frontier for RE community and industry setting.

The most common use of adaptive systems is in the previously mentioned areas of robotic and control theory, which demand dynamic readings of the context and immediate response to the system with as little human intervention as possible. The development of these systems has been significantly more challenging than the traditional model due to the need of mechanisms to automate and simplify the adaptation and modification of software after its installation [2]. Despite this, software engineers have focused their research on development of new technologies to manage the progressive complexity of software systems. The RE community and industry practitioners still lack templates and patterns to help and minimize the cost of developing such systems. It is noted in these circumstances the immense difficulty of specifying requirements for adaptive systems without previously defined and satisfactorily utilized pattern or tool.

Adaptive systems, as the name suggests, need to adapt to new context, but contextual uncertainties make it difficult to create, validate and manage the requirements. These systems are able to adjust their behavior at runtime as a response to the new reading of the context where the system is inserted [3]. However, despite being a very discussed topic in RE community, it still lacks tools and techniques to standardize its modeling.

RE technique and tools are satisfactory when the context is well known or evolves slowly. However, there is a need of mechanisms which automate and simplify the adaptation and modification of the system to operate in volatile contexts. The purpose of this research is to propose a specification process to adaptive systems focusing the definition of requirements that demand system adaptation. Such proposal is based on using Fuzzy Logic [4] and NFR-Framework [5].

Efforts to develop this research included a literature review on adaptive systems, requirements engineering, Fuzzy Logic and NFR-Framework. Such review aimed providing a theoretical basis for the definition of the object of the research that this study intends to produce. The activities began with a study about adaptive systems in general and about the works already produced by the RE community concerned to these systems. Papers and articles that dealt with these techniques and tool for specification and modeling of adaptive systems requirements were searched.

It was observed in the literature review that to manipulate requirements that go through changes at

 ISBN: 978-1-61208-304-9

runtime, studies with Fuzzy Set Theory could be helpful. Several articles related to the context of adaptive systems with set theory were researched. It was found that in the context of adaptive systems, it would be viable to approach Fuzzy Logic context [6][7][8][9], due to its use in problems involving fuzzy contexts.

Next, a model able to cover this complex context of requirements for adaptive systems was sought, opting for this NFR-Framework, which deals with uncertainties through the concepts of softgoals and represents them satisfactorily by means of SIG diagrams. The next step was to map the contexts explored, making a relationship among the three areas studied: adaptive systems, Fuzzy Logic, and NFR-Framework. To finish the relationship identified in the mapping, it was realized that the concept of requirements for adaptive systems should be better characterized. After this characterization, later called adaptive requirements, it was noted the need of creating a conceptual model. For the representation of such a model, a class diagram (from UML) was adopted, which shaped the main concepts involved, based on a previously done array of mapping.

The rest of this paper is organized as follows: an overview about adaptive systems and requirements for such systems are presented in section II; a proposal of requirements specification process for adaptive systems is presented in section III; a case study using the suggested proposal is reported in section IV; and conclusions and further works are presented in section V.

II. Adaptive Systems

Adaptive Systems are those that can be modified at runtime, due to changes in the system, in requirements or in the environment where they are implanted [3], depending upon various aspects, such as particular properties of a system, users requirements and characteristics of the environment.

According to Cheng [1], the simultaneous boom of information, the integration of technology and the continuous evolution of systems based on ultra large-scale software require new and innovative approach to building, implementing and managing software systems. To support this evolution, systems must become versatile, flexible, adapted to the three aspects mentioned above. To achieve this, the adaptive systems have become a topic of great interest in current researches in the Software Engineering Community [10].

There are requirements that are sensitive to the context in which the system will be implanted. Where the context is well known and static or evolutes slowly, the existing RE techniques can perform a good job. What is noticeable is that, increasingly, development projects are being challenged to build systems able to operate in volatile context, so that they are not totally previously understood [11][2].

Such systems must have the ability to dynamically adapt to new environmental context, but the contextual uncertainty that requires this adaptive potential hinders the elaboration, validation and management of its requirements and can be varied according to environmental requirements. The unexpected contexts may even lead to new requirements [3][12][13].

A. *Requirements for Adaptive Systems*

A conventional requirement (functional or non-functional) can be defined as a declaration of a service or constraint of a system being developed. It can also be simply defined as “something the client needs”. However, from the developer point of view, a requirement can also be defined as “something that needs to be developed”.

Developing adaptive systems demands making explicit the alternatives to achieve the goals, i.e., the variability in which and how it can be enhanced and the variability where and when, due to the operational environment.

This leads to the definition of requirements that are not only functional or non-functional, but also the specification of monitoring that takes under consideration the variability on an operational context, evaluation criteria and the behavior of alternative software being adopted by the software system at runtime to ensure the achievement of the user`s goals [14]. Requirements for adaptive systems are those that include the notion of variability associated to any functionality or a system quality constraint. Software requirements are generally characterized over the functional and non-functional classification. During the elicitation, the analyst first gives attention to the characterization of the stakeholders’ needs, which can be obtained through interviews or documents in a natural language. Requirements for adaptive systems reflect the uncertainties about the conditions at runtime due to the variability in the operational context and in the user`s necessities. In summary, adaptive systems are based on requirements that specify the necessity to modify the system behavior at runtime. Hereafter, requirements for adaptive systems with this characteristic are called adaptive requirements (AR).

III. PERSA: Requirements Specification Process for Adaptive Systems

This section presents the basic lines of the approach to the Requirements Specification Process for Adaptive Systems (PERSA – Portuguese acronym to *Processo de Especificação de Requisitos para Sistemas Adaptativos*). The process aims to aid the adaptive requirements specification activities through a well defined set of activities. Fuzzy Set Theory allows treating factors, such as ambiguity and uncertainty. Thus, the Fuzzy Sets, Fuzzy Logic and Fuzzy Reasoning provide the basis to generate the techniques to solve problems with a large applicability, especially in the control and decision making areas. In this work, the universe of fuzzy concepts formed by Fuzzy Set Theory, Fuzzy Logic and Fuzzy Reasoning will be mentioned as Fuzzy Logic. The NFR-Framework, which allows developers to work with the non-functional

 ISBN: 978-1-61208-304-9

requirements, systematically expressing and using them to guide the development process of software systems. The NFR-Framework has the softgoals as main component, which have a subjective nature.

PERSA process used Fuzzy Logic concepts as a basis for its development since they treat factors, such as ambiguity, uncertainty, and vague information in the solution of problems, enabling handling adaptive requirements, as well as NFR-Framework concepts, which has the definition of softgoals, fully compliant to the modeling of uncertain requirements, providing notation and semantics for the construction of SIG diagrams, which will be used as a graphic representation for adaptive requirements. The Fuzzy Logic concepts applied to PERSA process were entirely used and there was no expansion or alteration. The NFR-Framework concepts, also entirely used, will shape the process when building the SIG diagram and the adaptive requirements and not only the functional and non-functional requirements.

Thus, this work has begun with the challenge of creating an approach for adaptive system based on requirements (functional and non-functional), which may undergo variations during their lifespan. Requirements suffering variability, changes or extensions at runtime are classified as adaptive. Process aims to specify requirements for adaptive systems handling them with the Fuzzy Logic concepts and shaping them with NFR-Framework concepts.

The initial stage of requirements specification deals with the definition of global aspects of the project, determining items such as: project purpose, project scope and functional areas involved; goals to be achieved; technical and business assumptions that affects the project; critical factors for the success, among others. It is important to remember the necessity of being previously defined. This way, the activity of collecting functional and non-functional requirements must be performed in a conventional manner. The analyst may use any modeling technique available in the RE community. The PERSA Process begins its life cycle right after the stage of requirements collecting.

A. *Conceptualization*

As mentioned above, adaptive requirements (AR) are those which include the notion of variability associated with any functionality or with any quality constraint of the system [15]. The first step in the creation of PERSA process consisted of the attributes identification for each concept related to an adaptive system concept:

User`s goals: what the software must meet. The user`s goals must be achieved.

Environment Variability: the environmental context where the software is implanted can change.

Alternative Behavior: according to a new reading of the environmental behavior, the behavior of the software may change.

Mutant Variables: are those which do not offer a clear definition of all values they may take. For example, the variable “fire intensity” may have values like high, middle or low.

Evaluation Criteria: an analysis of the software is performed after a change to check it is still meets the user`s goals satisfactorily.

Below the list of attributes of Fuzzy logic:

Linguistic Variables: have values with names of Fuzzy Sets. They can be put in a specific language, from primary terms, logic connectives, modifiers or delimiters.

Membership Functions: each Fuzzy Set is characterized by the membership function.

Fuzzification Interface: identifies the input variables values, which characterize the state of the system which normalizes it in a universe of standardized speech.

Inference Rules: represent the model of the system to be controlled. They characterize the goals and the control strategy used by specialists.

Defuzzification Interface: consists in obtaining a single discrete value usable in a concrete action of controlling the real world from the obtained fuzzy output values.

The list of NFR-Framework attributes completes the group of concepts in which PERSA is based on:

Softgoals: represent and aid developers to work on non-functional requirements (NFR).

SIG Diagram: the representation and use of *NFR-Framework* are made through SIG Diagrams.

Evaluation: determines the degree of satisfaction of the *softgoal* in its dependency relation with others.

Contribution: type of positive or negative collaboration to achieve the goals.

Interdependencies: are inter-relations between the softgoals refinements aiming the satisfaction of the related softgoals.

Catalogues: store the acquired knowledge structuring and enabling the reuse.

B. *PERSA Process Activities*

As previously reported, PERSA process starts right after the requirements survey ends. The PERSA process activities concerned with the creation of fuzzy rules was based on Mamdani method, which is a well known method to specify fuzzy rules. The PERSA process activities were organized in three main phases:

1st phase: Analysis of the Requirements List.

2nd phase: Fuzzy Modeling:

1st Stage: Create Linguistic Variables.

2nd Stage: Create Fuzzy Sets.

3rd Stage: Add values to the Fuzzy Sets.

4th Stage: *Fuzzification* Process:

1. Charge Input Values;
2. Choose Membership Function;
3. Perform Calculations according to Membership Functions;
4. Assemble *Fuzzification* Matrix.

5th Stage: Assemble Inference Rules:

1. Use *Fuzzification* Matrix;

2. Seek Specialist in Business Rule;
3. Choose Mamdani Method;
4. Build Knowledge Base according to Mamdani Method;
5. Interview Specialist;
6. Add data to the Knowledge Base;
7. Calculate Function MINIMUM;
8. Generate Graphic of Inference Rules;
9. Calculate Function MAXIMUM;
10. Generate Knowledge Base Graphic.

6th Stage: *Defuzzification* Process*:*

1. Use Knowledge Base Graphic;
2. Choose D*efuzzification* Method;
3. Use points from the Graphic of Knowledge Base;
4. Make Calculations.

3rd Phase: NFR Modeling

1st Stage: Specify Goals.

2nd Stage: Name NFR *Softgoals*.

3rd Stage: Generate SIG Diagram:

1. Create NFR *Softgoals*;
2. Decompose *Softgoals;*
3. Verify Operationalization;
4. Verify Decomposition;
5. Verify Correlation;
6. Select Operationalizations.

The input to PERSA process comes from the requirements elicitation performed in a conventional way. The elicited requirements are analyzed with the intention to find those that present variations during the adaptive system life cycle. That is the first phase of the process. For each requirement that presents meaningful variability, the 2nd and 3rd phases of PERSA process must be performed. At the 2nd phase a fuzzy model is created following the steps listed before. At the 3rd phase a NFR model is created, associating linguistic variables and fuzzy sets to the softgoals. The fuzzy model and NFR model complement each other, helping requirements engineers to better understand the adaptive requirements.

IV. CASE STUDY

In this section a case study is presented, in which PERSA process was used integrally, aiming to specify adaptive requirements in the analyzed problems. The case study, called "cook`s problem", consisted of the specification of an automate system to prepare steaks, requiring an adaptive system related to the different types of meat, which are prepared according to the customers` order being rare, medium or well-done.

A. Cook`s Problem

As recommended by PERSA process, the input variables, the output and their respective fuzzy sets were initially defined, as showed in Tables I and II. In Figure 1, the graphics with the values of fuzzy sets of the variables "Time" are presented. The horizontal axis represents membership degrees and vertical axis represents the fuzzy sets thresholds.

TABLE I. VALUES RANGE OF THE INPUT FUZZY SETS

	Reddish			Pinkish			Brown		
	Beginning	Average	End	Beginning	Average	End	Beginning	Average	End
Tone	155	205	235	95	125	175	55	75	135
Time	0.5	1	2	1	2	3	2.5	3	4.5
	Short			Average			long		
	Beginning	Average	End	Beginning	Average	End	Beginning	Average	End

TABLE II. VALUES RANGE OF THE OUTPUT FUZZY SETS

2.	Output Variables														
	Rare			Medium			Well Done			Raw			Burnt		
	Beginning	Average	End	Beginning	Average	End	Beginning	Average	End	Beginning	Average	End	Beginning	Average	End
State	1	2.5	4	3	4.5	7	6	8.5	10	-5	-2	1	10	12	15

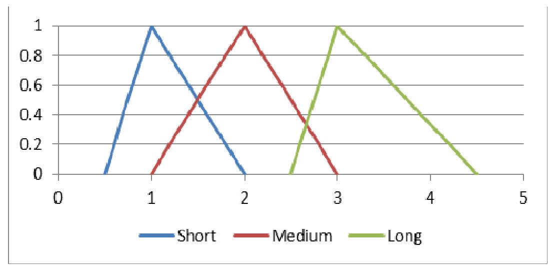

Figure 1. Input Variable "Time" with the values of Fuzzy Set

According to the fuzzy sets, the membership function triangular was chosen. In this case study, the system was fed with the values 98 for the input variable Tone and 1.9 for the input variable Time.

TABLE III. FUZZIFICATION MATRIX

Input Variable	**Input Value**	**Fuzzy Sets**		
Time	1.9	*µShort*	*µMedium*	*µLong*
		0.10	0.90	0.00
Tone	98	*µReddish*	*µPink*	*µBrown*
		0.00	0.10	0.62

The fuzzification matrix was made from the result of the Membership Function *Triangular*, according to Table III. With this done, it moved to the fifth stage of the second phase of PERSA process. At this point, the process requires a specialist to assist the definition of the system inference rules.

This fifth stage of the second phase may be considered essential since it contains the main difference between the adaptive and the conventional system. Here, the table Knowledge Base is constructed, based on Mamdani method, when the specialist determines the results of each combination among the input variables. In the cook`s problem case study, according to Table IV, it may be noted that the specialist`s answers are in the last column. For example, *If Short Time and Reddish Tone, then state of the steak = Raw*.

 ISBN: 978-1-61208-304-9

TABLE IV. KNOWLEDGE BASE MATRIX – COMPLETE

Reg.	Time		Tone		State	Minim.
	Fuzzy	Pert.	*Fuzzy Set*	Pert.		
01	*µShort*	*0.10*	*µReddish*	*0.00*	Raw	**0.00**
02	*µShort*	*0.10*	*µPink*	*0.10*	Rare	**0.10**
03	*µShort*	*0.10*	*µBrown*	*0.62*	Medium	**0.10**
04	*µAverage*	*0.90*	*µReddish*	*0.00*	Rare	**0.00**
05	*µAverage*	*0.90*	*µPink*	*0.10*	Medium	**0.10**
06	*µAverage*	*0.90*	*µBrown*	*0.62*	Well Done	**0.62**
07	*µLong*	*0.00*	*µReddish*	*0.00*	Medium	**0.00**
08	*µLong*	*0.00*	*µPink*	*0.10*	Well Done	**0.00**
09	*µLong*	*0.00*	*µBrown*	*0.62*	Burnt	**0.00**

To each rule created by the Mamdani Method and described in Table II, the function Minimum must be calculated and the graphics must be generated, which are the basis to create the Knowledge Base Graphic, illustrated in Figure 2. Through this, the Centroid is calculated and the mathematical data are transformed in numbers from the real world. In the case study, the inputs inform that the steak contains 56% (fifty six percent) of characteristics in the Fuzzy Set "Well Done" and thus, the fuzzification process is finished in PERSA Process.

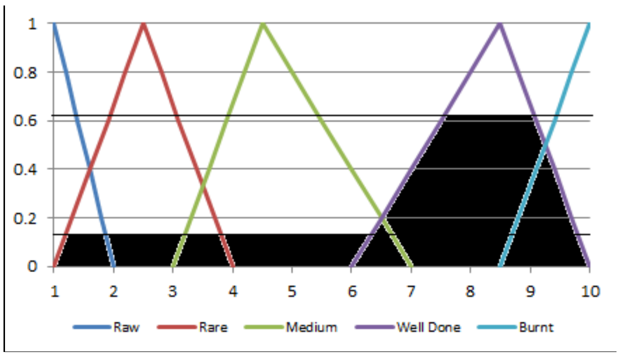

Figure 2. Knowledge Base Graphic

TABLE V. DEFUZZIFICATION METHOD ADOPTED IN THE CASE STUDY - CENTROID CALCULATION

Score in the Graphic	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
Rare	0.1	0.1	0.1						
Medium				0.1	0.1	0.1			
Well Done						0.2	0.6	0.7	0.7
MAXIMUM	0.1	0.1	0.1	0.1	0.1	0.2	0.6	0.7	0.7
*Score * MAXIMUM*	0.15	0.25	0.35	0.45	0.55	1.3	4.5	5.95	6.65
SUM (MAX)	**2.7**								
SUM (Score * MAX)	**20.15**								
RESULTADO	**7.462962963**								

The result of defuzzification method presented 7.46 in the output variable, showed in Table III. This means that with the inputs in the system (time = 1.9 min. and tone = 98), this Steak contains 56% (fifty six percent) of characteristics inside the Fuzzy Set "Well Done" and 0% (zero percent) membership in the other sets. Then, it can be said that the steak is "well done".

The last phase of the PERSA Process, named NFR Modeling, generates SIG diagrams: to each input variable a NFR softgoal is created, as illustrated in Figure 3. It may be noted that the main difference between modeling in a conventional system and an adaptive one, through NFR Modeling is in Figure 3, exactly in the "Verify Inference Rules" softgoal. To meet this, three conditions must be met:

- The "Monitoring Color" softgoal must be Pink;
- The "Monitoring Time" softgoal must be Short or Average Time;

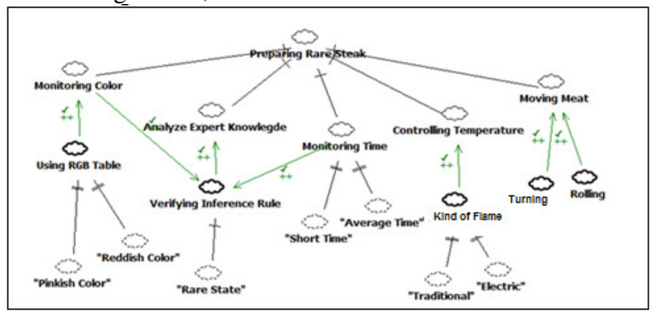

Figure 3. SIG Diagram specifying the adaptive requirement "Prepare Rare Steak"

If the two softgoals above were satisfied, the "Verify Inference Rules" softgoal must be met with the "Rare State" claim softgoal.

B. Discussion and Analysis of Results

With the purpose of observing and validating the activities suggested in PERSA process, the theoretical proposal was applied in a case study, which contemplates an adaptive system aiming to determine the degree of understanding, the clarity of activities and the necessary adjustments to improve the activities proposed in PERSA process. PERSA Process was divided in three different stages: analyze the list of requirements, treat adaptive requirements through fuzzy modeling, modeling adaptive requirements through NFR modeling. In the implementation of adaptive requirements specified in the case study, the table Knowledge Base (Table IV) must modify itself at runtime to satisfy the main goal of the adaptive systems, which consists the possibility of alterations at runtime due to the variability in the environmental context.

In the Cook`s Problem, it can be imagined a reading of the tone "Black": in case it does not fit in any of the inference rules and that would, by approximation lead the adaptive system to an adjustment to this situation by creating a new rule bases on a preexistent one, similar to the color "Black", thus continuing its running. The new rule would have the following definition: *If Short time and Black Tone, the state of the Steak Burnt*. It is emphasized that the column filled by the specialist do not alter, only the columns with the fuzzy sets. At the end of the case study explanation, there is a satisfactory assessment, because it reached its purpose of specifying requirements for adaptive system. PERSA process specifies adaptive requirements clearly and systematically. Though it is a support technique to software specification demanding the Requirements Engineering to acquire knowledge about Fuzzy Logics and NFR-Framework, it leads to improvements in quality and productivity when developing adaptive systems which justifies the cost of initial investment for the learning of the process. In conclusion, despite performing only few case studies and the need of a wider range of evaluation, based on this initial assessment, the specification outcome is positive, achiever of its goal, confirming that PERSA Process specifies requirements for adaptive systems clearly, effectively and systematically

V. CONCLUSION

This paper presented an approach of requirements specification for adaptive systems, based on the characteristics identified in systemic context with high variability and many fuzzy variables, full of uncertainties as well as the relevant definitions to the adaptive requirements modeling, based on Fuzzy Logic and *NFR-Framework.*

The purpose of this research aimed to assist the existing lack in the requirements specification for adaptive systems. The requirements specification for any type of system is not a trivial task, since it still presents problems identified decades ago. Thus, the adaptive requirements specification, which has special features, such as the possibility of modifying at runtime, makes the challenge even greater.

A. Main Contributions

PERSA Process presented in this study aimed to recommend a systematic way to the activities of requirements specification for adaptive systems. The following aspects may be indicated as this study`s main contributions:

- The conception of a requirements specifications process for adaptive systems;
- The creation of a specific requirements documentation for adaptive systems;
- The specification of systemic uncertain and with vague information contexts.

This study limits itself to the requirements specification for adaptive systems by PERSA process. Slightly extending beyond limitation and crossing the border with code/implementation phase, it may be stated that the core of adaptive system is in the creation and management of the Knowledge Base Matrix (as seen in Tables IV and V). The Knowledge Base should be modified at runtime to satisfy the changes in the environmental context, being the main difference of an adaptive system and a conventional one.

B. Future Works

This work, through a series of new proposals, can be expanded by further studies. To this end, the following proposals are highlighted:

- Adjustment and inclusion of activities in PERSA process identified by the study of more complex cases;
- Validation of the proposed PERSA process by developing other case studies;
- Development of an automated tool to support and facilitate the use of PERSA process;
- Creation of a repository for storing and retrieving the generated artifacts along the use of PERSA process;
- To perform the next phase of Requirements Engineering (validation), based on the artifact generated by PERSA Process.

REFERENCES

[1] B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, “Software Engineering for Self-Adaptive Systems: A Research Roadmap”, In: Software Engineering for Self-Adaptive Systems, Springer, 2009, pp.1-28.

[2] A. J. Ramires, B. H. Cheng, and P. K. Mckinley, “Adaptive Monitoring of Software Requirements”, In: Requirements@Run.Time, First International Workshop on, 2010, pp. 41-50.

[3] G. Brown, B. H. Cheng, H. Goldsby, and J. Zhang, “Goal-oriented Specification of Adaptation Requirements Engineering in Adaptive Systems”, In: SEAMS '06: Proceedings of the International ICSE Workshop on Self-adaptation and self-managing systems. ACM, 2006, pp. 23–29.

[4] L. A. Zadeh and R. R. Yage, Fuzzy Sets and Applications: Selected Papers by L.A. ZADEH, Wiley-Interscience, 1987

[5] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, “Non-Functional Requirements in Software Engineering” In: The Kluwer International Series in Software Engineering, Vol. 5, 1999.

[6] L. Baresi, L. Pasquale, and P. Spoletini, “Fuzzy Goals for Requirements-driven Adaptation”, In: 18th IEEE International Requirements Engineering Conference, 2010, pp.125-134.

[7] G. Klir, U. H. St. Clair, and B. Yuan, Fuzzy Set Theory – Foundations and Applications, United States : ed. Prentice Hall, 1987.

[8] W. Pedrycz and F. Gomide, An Introduction to Fuzzy Sets: Analysis and Design, A Bradford Book, 1998.

[9] M. Serrano and J. C. S. P. Leite, “Dealing with softgoals at runtime: A fuzzy logic approach”, In: Requirements@Run.Time, 2nd International Workshop on, 2011, pp. 23 - 31.

[10] S. Liaskos, S. A. Mcllraith, S. Sohradi, and J. Mylopoulos, “Integrating Preferences into Goal Models for Requirements Engineering”, In: 18h IEEE International Requirements Engineering Conference, 2010, pp. 135-144.

[11] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, and A. Finkelstein, “Requirements-Aware Systems: A research agenda for RE for self-adaptive systems”, In: 18th IEEE International Requirements Engineering Conference, 2010, pp. 95-103.

[12] N. A. Qureshi and A Perini, “Requirements Engineering for Adaptive Service Based Applications”, In: 18th IEEE International Requirements Engineering Conference, 2010, pp.108-111.

[13] J. Pimentel and J. Castro, “Specification of Failure-Handling Requirements as Policy Rules on Self-Adaptive Systems”, In: 14th Workshop on Requirements Engineering, 2011. pp. 345 -356.

[14] N. A. Qureshi and A. Perini, “Engineering Adaptive Requirements”, In: SEAMS '09: Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems. ACM, 2009, pp. 126-131.

[15] N. A. Qureshi, S. Liaskos, and A. Perini, “Reasoning about adaptive requirements for self-adaptive systems at runtime”, In: Requirements@Run.Time, 2nd International Workshop on, 2011, pp. 16 – 22.