A Fuzzy Service Adaptation Based on QoS Satisfaction

Barbara Pernici and Seyed Hossein Siadat

Politecnico di Milano
Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
{pernici,siadat}@elet.polimi.it

Abstract. Quality of Service (QoS) once defined in a contract between two parties may change during the life-cycle of Service-Based Applications (SBAs). Changes could be due to system failures or evolution of quality requirements from the involved parties. Therefore, Web Services need to be able to adapt dynamically to respond to such changes. Adaptation and evolution of services are playing an important task in this domain. An essential issue to be addressed is how to efficiently select an adaptation while, there exists different strategies. We propose a fuzzy service adaptation approach that works based on the degree of QoS satisfaction. In particular, we define fuzzy parameters for the QoS property descriptions of Web Services. This way, partial satisfaction of parameters is allowed through measuring imprecise requirements. The QoS satisfaction degree is measured using membership functions provided for each parameter. Experimental results show the effectiveness of the fuzzy approach using the satisfaction degree in selecting the best adaptation strategy.

Keywords: QoS, service adaptation and evolution, fuzzy logic.

1 Introduction

In Service-Based Applications (SBAs), Quality of Service (QoS) parameters may change during the life cycle of the application. Web service adaptation is an important phase to deal with such changes. Handling changes in a demanding and adaptive environment is a vital task. One main issue lies in QoS property descriptions of Web Services. This involves specifying service requirements in a formal way, monitoring and dynamically adapting and evolving the services with respect to the QoS changes. Static adaptation is impractical due to the changing environment and high cost of maintenance and development. Specifying all possible alternative behaviour for adaptation at design time is impossible. Therefore, a declarative approach is required at run-time to support adaptation decisions.

In order to perform run-time decisions for adaptation in a volatile environment, one issue is to consider the imprecise evaluation of QoS properties. Existing approaches do not allow partial satisfaction of parameters. It is required that

H. Mouratidis and C. Rolland (Eds.): CAiSE 2011, LNCS 6741, pp. 48–61, 2011.

services should be able to tolerate a range of violation in their quality description. However, handling this toleration need to be done with special care. An important issue to address in SBAs is to what extent the QoS parameters of a Web Service are satisfiable. The answer to this issue could be a basic for making adaptation decisions. However, this issue has not been addressed adequately in the literatures. Evaluating the extent of parameter satisfaction is necessary to help the selection of best adaptation strategy.

As an initial step to this, in [3] we provided conditions under which QoS changes are acceptable. We used a temporal logic namely Allen's Interval Algebra (AIA) [2] to formally specify the non-functional properties of web services. We then used the AIA to reason about changes of quality parameters and their evolution. In this paper, we extend [3] and propose a fuzzy approach to support service adaptation and evolution. We define *fuzzy parameters* for QoS property description of Web Services. Fuzzy parameters could be considered as fuzzy sets and measured based on their value of membership. Satisfaction degree of fuzzy parameters is measured according to their actual distance of the agreed quality ranges in the contract. The goal of this paper is to provide flexibility for service specification by applying fuzzy parameters. Using a fuzzy approach allows us to deal with reasoning on the quality violations that is approximate rather than accurate. At the end, we propose different categories of adaptation that perform based on the satisfaction degree. Experimental results show the effectiveness of using the fuzzy approach over the non-fuzzy one in making decisions for adaptation.

The remainder of the paper is structured as follows. Section 2 describes the major related work. In Section 3 we present a definition for QoS property description of services through introducing fuzzy parameters. In Section 4 we specify satisfaction functions for each parameter to measure to what extend the QoS is achieved with respect to the existing contract. We explain the decision making mechanism in Section 5 that works based on the satisfaction degree. Section 6 provides experimental result using a simulator and evaluates the effectiveness of the proposed approach. Section 7 concludes the paper and discusses our future work.

2 Related Work

Deviation of quality ranges from the existing contract may produce a system failure and bring dissatisfaction for customers. To this end, the evolution and adaptation of web services are becoming two important issues in reacting to the various changes in order to provide the agreed QoS stated in the contract. Recently, many adaptation strategies and methods have been proposed in the literature. However, most of the work in service adaptation concentrates on the technical issues and definition of mechanisms for adaptation rather than considering QoS perspective. A list of adaptation strategies for repair processes in SBAs is provided in [6] and [1]. For example, [7] proposed a service replacement approach for adaptive Web Service composition and execution, while Canfora et al. [5] presented a re-composition approach dealing QoS replanning issues at run time using late binding technique. However, none of these works consider the consequences and potential overheads of adaptation. To this end, for example, an environment for compensation of Web Service transactions is proposed in [25]. In order to consider the overall value of a change, [15] presented an approach called *value of changed information* (VOC). Furthermore, an adaptation mechanism is proposed based on VOC in [8]. However, these works have the limitation that they do not take into account the satisfaction level of services. Making adaptation decisions and evaluating them is therefore complicated and has consequences that are often neglected. Some qualitative and quantitative techniques has been proposed, however evaluating impacts of adaptation still remains as an open challenge.

One core issue to address is the definition of a flexible description for Web Services. Formulation of service specifications/requirements has been studied in the literature. In autonomic systems and in particular web services, reasoning about such specification is a hard job due to the changing environment that affects service requirements. Although a lot of research has been conducted for functional Web Service description, only a few efforts have been done with respect to non-functional properties description of Web Service. Among syntactic and semantic WS description we refer to the work done in [30], [21] and [16] which they also provided algorithms for service selection based on such description. A major limitation of those papers and other similar ones is due to not considering the partial satisfaction of the QoS attributes. With this regards, [23] provided a semantic Policy Centered Meta-model (PCM) approach for non-functional property description of web services. A number of operators (e.g. greaterEqual, atLeast) for numeric values are defined in the model for determining tradeoffs between various requests. Therefore, the approach can support the selection of Web Services that partially satisfy user constraints. In [20] and [19], the authors extend the approach proposed in [23] by proving a solution for Web Service evaluation based on constraint satisfaction problem. The approach uses utility function to present the level of preferences for each value ranges defined in the service description. However, it does not take care of adaptation issues and controlling values at run-time. In [22], the authors discuss about fixable and non-fixable properties to deal with bounded uncertainty issue. Constraint programming is used as a solution, however, there is no evaluation of the work.

It is required to provide a framework to evaluate alternatives and quantify their impact for making decision decisions. Each alternative has different degree of satisfaction and their impact has to be evaluated in order to select the best adaptation strategy. A quantitative approach applying a probabilistic modeling is used for partial goal satisfaction in [18]. Dealing with the uncertainty issue is one major problem in order to formulate and manage service specification. Thus, recently researchers are investigating to incorporate this uncertainty into the service specification. In [14], the author provides support for reasoning about uncertainty. A goal-base approach for requirement modeling in adaptive systems is proposed in [9] which uncertainty of the environment is taken into account.

Furthermore, a language named RELAX is developed for specifying requirements in adaptive systems [26,27] in which certain requirements could be temporarily relaxed in favor of others. In general, different temporal logics have been used for formal specification of requirement. Linear Temporal Logic (LTL) has been used in [4,11] to formally specify requirements in a goal oriented approach. In particular, LTL is extended in [31] and named A-LTL to support adaptive program semantics by introducing an adaptation operator. [3] uses Allen's interval algebra for the formal specification of service requirement. Those approaches have limitations such that they are unable to consider environmental uncertainty and behave in a binary satisfaction manner.

Fuzzy approach [29] is an alternative to concur such limitations of aforementioned approaches. However, the fuzzy approach may not be the only alternative to deal with uncertainty. Different mathematical and frameworks are presented in the literature to address the uncertainty issue and partial satisfaction of the requirements. For example, making decisions about non-functional properties using Bayesian networks is proposed in [13] while [17] used a probabilistic method for this purpose. Applying fuzzy logic to incorporate uncertainty and making decisions has been proposed in other domains such as management, economy and many aspects of computer science, however, to the best of our knowledge there is very little of such application in adaptation of web services. As of such, [10] proposed a fuzzy approach for assigning fitness degrees to service policies in a context-aware mobile computing middleware. A trade-off analysis using fuzzy approach for addressing conflicts using imprecise requirements in proposed in [28]. With respect to partial satisfaction of requirements, [12] provided a web service selection approach using imprecise QoS constraints.

There are several different approaches towards adaptation of web services. This diversity yields from a missing consensus on the required decision making to automatically perform web service adaptation. Therefore, in this paper we propose a fuzzy adaptation approach as a possible way in providing a foundation of such a consensus which is based on the satisfaction degree of QoS parameters.

3 Fuzzy Parameters for QoS Property Description

This section is devoted to present a formal definition of quality *parameters* in a service description and is concerned with QoS property descriptions of Web Services. The formal specification we propose has been inspired and is an extension of our previous work [3]. We extended the work by defining *fuzzy parameters* for such service description. Fuzzy parameters could be considered as fuzzy sets and measured based on their value of membership. Satisfaction degree of fuzzy parameters is measured according to their actual distance of the agreed quality ranges in the contract. Having introduced the fuzzy parameters it is possible to understand to what extent the quality parameters are violated/satisfied. This way, partial satisfaction of parameters is allowed through measuring imprecise requirements.

We define set $\mathcal{D}$ to contain the quality dimensions (such as availability, execution time, price or throughput) identified and agreed by the service provider and consumer. Each quality dimension has a domain and range; e.g., availability is a probability usually expressed as a percentage in the range 0-100% and execution time is in the domain of real numbers in the range $0..+\infty$. A quality dimension d can be considered *monotonic* (denoted by d^+) or *antitonic* (d^-); monotonicity indicates that values closer to the upper bound of the range are considered better, whilst with antitonic dimensions values closer to the lower bound are considered better. A *parameter* m associates a quality dimension to a value range [3].

If a parameter is non-fuzzy (strict) its satisfaction degree will be evaluated in a binary manner (Yes or No). In contrast, fuzzy parameters (relaxed) will be evaluated in a fuzzy manner which shows different degree of satisfaction ($x \in [0,1]$). Note that we also provide value ranges for both parameters regardless of being fuzzy or non-fuzzy. The satisfaction degree of fuzzy parameters will be measured using *membership functions* provided for each parameter. In the following we provide the extended definition of a *parameter* based on the definition introduced in [3]. In particular, we define a type t for a parameter that can be either strict or fuzzy.

Definition 1 (Parameter). *We define a Parameter $m \in \mathcal{M}$ as a tuple $m := (d, v, t), d \in \mathcal{D}, v \in \mathcal{V}, t \in \{s, f\}$. where $\mathcal{D}$ is the set of quality dimensions, $\mathcal{V}$ is the set of ranges for all quality dimensions $\mathcal{D}$, s represent a strict parameter and f represent a fuzzy parameter.*

QoS once defined in a contract between two parties may change during a service life-cycle. Changes could be due to system failures or evolution of quality requirements from the involved parties. Therefore, Web Services need to be able to adapt dynamically to respond to such changes. Adaptation and evolution of services are playing an important task in this domain. However, adaptation of web services needs to be performed in an appropriate manner to accommodate QoS changes/violations by choosing the best adaptation strategy. Defining service description with the proposed fuzzy parameters provides a flexible situation in dealing with adaptation decisions. We discuss how it can facilitate the adaptation of web services through an example. According to the new definition of parameters, we consider availability and response time as fuzzy parameters. Let us assume an example of a contract with initial value ranges of availability between 80% to 90% and response time between 2 to 5 seconds. We use this example throughout the paper.

We provided situations in which new QoS ranges could be still acceptable for both parties according to the existing contract [3]. We introduced a compatibility mechanism that used parameter subtyping relation and Allen's Interval Algebra [2] for comparing value ranges and their evolution. The provider and requestor are compatible with each other according to the existing contract if the QoS changes are in one of the acceptable situations. If the compatibility is not provided, however it does not give any information about the degree of satisfaction/dissatisfaction of the offered service. For example if the new range

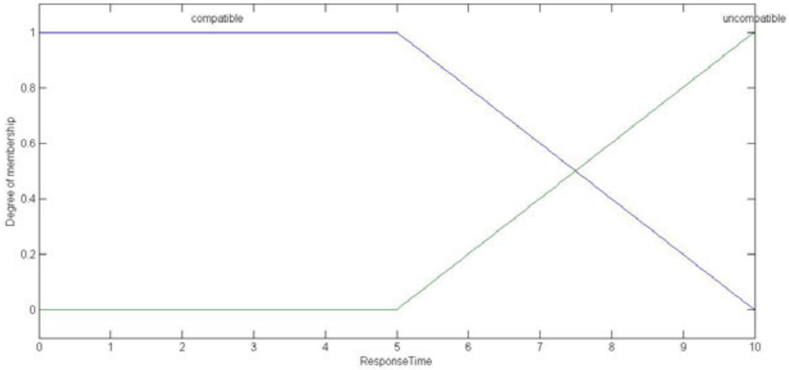

Fig. 1. Membership functions for response time

of availability is less than 80%, this is not considered as an acceptable situation and it is considered as a violation. In such cases, we would also like to understand to what extent the quality parameter and the aggregated service quality are satisfactory. An availability of 75% might still be acceptable if we consider the partial satisfaction of quality ranges.

4 Specifying Satisfaction Function

Having defined the fuzzy parameters we are able to apply the fuzzy logic. As for the first step we need to know the right amount of quality satisfaction. Previously in [3], we provided a compatibility mechanism to understand under which conditions the changes are acceptable. The approach suffers from the limitation that changes are considered either compatible or incompatible with the contract. This means, quality changes are calculated in a binary approach which it does not take into account clearly the relation of quality parameters with their satisfaction. To say it in other way, the QoS parameters are measured in a precise manner and their partial satisfaction is not taken into account. In the following we provide mechanisms to allow partial satisfaction of quality parameters imprecisely using fuzzy sets.

The main point of using fuzzy logic is to find a relation and to map our input space to the output space. The inputs here are namely service availability and response time and the output is the overall satisfaction degree of them. For each QoS parameter in the service description we provide a membership function that represent the level of satisfaction of each parameter. The membership functions map the value of each parameter to a membership value between 0 and 1. We use a piece-wise linear function, named *trapezoidal* membership function, for this purpose. Membership functions for ResponseTime and availability are shown in Figures 1 and 2. Both membership functions have two linguistic states namely *compatible* and *incompatible* and they are identified according to the initial value ranges of the contract. Figure 1 shows that the response time of 0 to 5 has the maximum degree of compatibility; however, the membership degree decreases while the response time increases. It also shows that response time has the minimum degree of incompatibility between 0 to 5 seconds; however, the membership degree increases while the response time increases. Note that the response time is set to be 2 to 5 seconds in the contract; however, the range between 0 to 2 is also acceptable with the same membership value as the initial range in the contract has [3]. With the same reasoning, the availability of 80 to 100 has the maximum degree of compatibility illustrated in Figure 2. However, the membership degree of compatibility decreases while the availability decreases.

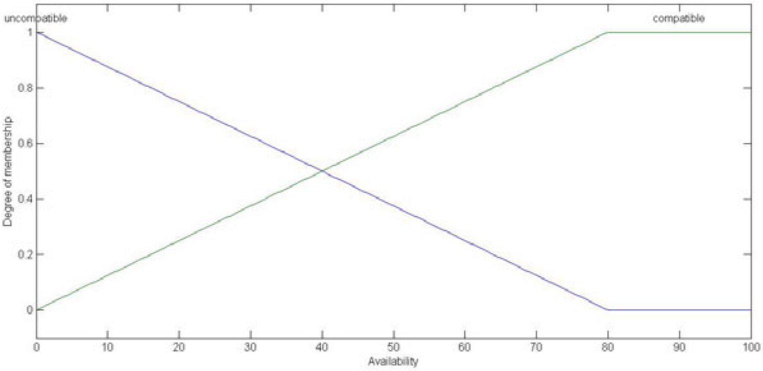

Fig. 2. Membership functions for ResponseTime and Availability

Having defined the membership functions, the mapping between the input and output space will be done by defining a list of *if-then statements* called *rules*. We have already defined what do we mean being compatible and incompatible for the quality parameters and specified their ranges using membership functions. Since we are relaxing the antecedent using a fuzzy statement, it is also required to represent the membership degree of the output (i.e. here satisfaction). Therefore, the satisfaction degree is also represented as fuzzy sets: satisfaction is low, satisfaction is average and satisfaction is high.

We define three if-then rules as below. it represents the *antecedent* and *consequent* of the rule. All the rules are applied in parallel and their order in unimportant. We define the fuzzy union/disjunction (OR) and the fuzzy conjunction/intersection (AND) using *max* and *min* functions respectively. Therefore $A\,AND\,B$ is represented as $min(A, B)$ and $A\,OR\,B$ is represented as $max(A, B)$.

1. $If\ (ResponseTime\,is\,compatible)$ *and* $(Availability\,is\,Compatible)$ *then* $(Satisfaction\,is\,high)$.

2. $If\ (ResponseTime\,is\,incompatible)$ *or* $(Availability\,is\,incompatible)$ *then* $(Satisfaction\,is\,average)$.

3. $If\ (ResponseTime\,is\,incompatible)$ *and* $(Availabilityis\,incompatible)$ *then* $(Satisfaction\,is\,low)$.

5 Decision Making for Adaptation and Evolution

We use the satisfaction degree calculated using the fuzzy inference system for the adaptation and evolution decision making. The decision making mechanism works based on the algorithm we provided in [24]. The algorithm evaluates the evolution of the service and decides which adaptation strategy to take with respect to the predefined threshold degree for QoS satisfaction. The two main decisions are the *internal renegotiation* in which the changes are compatible with the service description in the contract and *service replacement* in which the changes are incompatible with the existing contract. The former case deals with the internal contract modification with the same provider and requester while the earlier case requires the selection of a new service and establishment of a new contract which can result in a huge loss of time and money.

Having provided such a decision making mechanism allows us to offer a flexible adaptation mechanism. This is done by identifying threshold to what constitutes *compatible* and *incompatible*. Using satisfaction degree allows us to define the criticality of a change/violation. Therefore, we are able to understand whether a violation is critical and it results in a service replacement or the violation is still acceptable. This way, a slight change from the quality ranges defined in the contract will not trigger the adaptation. Table 1 shows the result of checking for compatibility for a possible set of changes. The comparison is between our fuzzy approach and a traditional non-fuzzy one that works based on the precise evaluation of the quality ranges in the contract.

Table 1. Comparing the adaptation decisions using fuzzy and non-fuzzy approach

Change	**Replacement?** **(Non-fuzzy/Fuzzy)**	**Change**	**Replacement?** **(Non-fuzzy/Fuzzy)**
$S_1 = (6, .90)$	Yes/No	$S_2 = (7, .75)$	Yes/Yes
$S_3 = (5, .85)$	No/No	$S_4 = (3, .70)$	Yes/No
$S_5 = (2, .85)$	No/No	$S_6 = (2, .78)$	Yes/No
$S_7 = (2, .60)$	Yes/Yes	$S_8 = (3, .90)$	No/No
$S_9 = (7, .95)$	Yes/Yes	$S_{10} = (6, .50)$	Yes/Yes

In the fuzzy approach the replacement is based on the satisfaction degree. However, in the non-fuzzy approach a service replacement is necessary if any parameters are violated from the initial range, albeit minor deviation. For example in $S_1 = (6, .90)$, changing the response-time to 6s will not result a service replacement applying the fuzzy approach since it has the satisfaction degree of almost 83%. While applying a non-fuzzy approach, it is considered a violation because it does not respect the initial response-time range (2, 5) in the contract. However, if a change results in a low satisfaction degree, service replacement is necessary in both approaches as in the case $S_{10} = (6, .50)$ which the satisfaction degree is around 62%.

6 Experiments and Implementation

Having defined the membership functions and rules in the previous sections, we have built and simulated a fuzzy inference system to interpret rules. The process has different steps including: fuzzification of input quality parameters, applying fuzzy operators to the antecedent, implication from the antecedent to the consequent, aggregation of the results for each rule, and defuzzification. A view of the simulator including the previous steps is illustrated in Figure 3 in which a complete fuzzy inference system is represented.

The first step is to apply the membership functions to map each QoS parameters to the appropriate fuzzy set (between 0 and 1). We used two inputs of Availability (interval between 0 to 100) and Response-time (interval between 0 to 10). The inputs are mapped to fuzzy linguistic sets: availability is compatible, availability is incompatible, response-time is compatible, and response-time is incompatible. Figures 1 and 2 show to what extent the availability and response-time are compatible. The next step is to give the result of the fuzzified input parameters to the fuzzy operators. According to the rules, AND and OR operators are applicable. This will give us a degree of support for each rule. Next is applying the implication method that uses the degree of support to calculate the output fuzzy set. We used a *minimum* method to truncate the output fuzzy set for all the rules separately. However, we apply all the rules in parallel and we do not define any priority and weight for them.

At the end of the implication, we apply an aggregation method to combine all the rules. This way, the outputs of each rule represented in fuzzy sets are combined into a single fuzzy set. A maximum method is used for the aggregation. The last step is to defuzzify the fuzzy set resulted after the aggregation step. We applied a *centroid* method to calculate the defuzzification process. The method returns the center of the area under the curve. Figure 3 shows that the

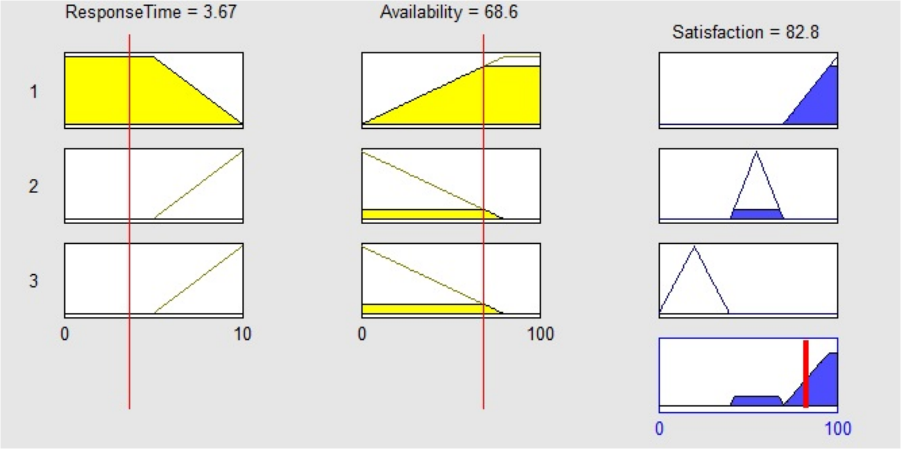

Fig. 3. A view of the simulator for fuzzy inference system

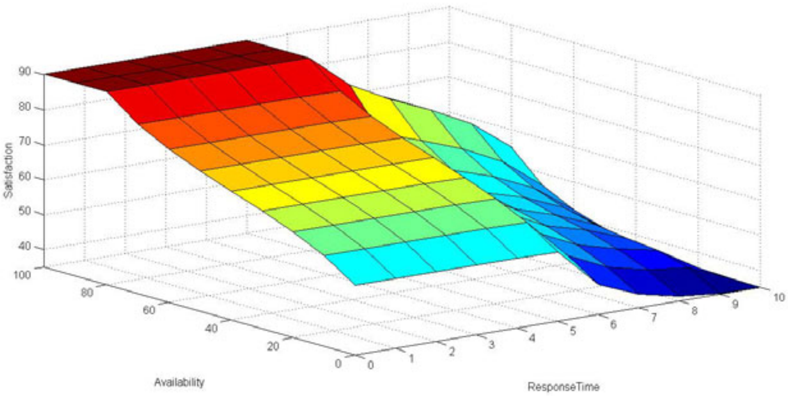

Fig. 4. The output of satisfaction degree according to ResponseTime and Availability membership function

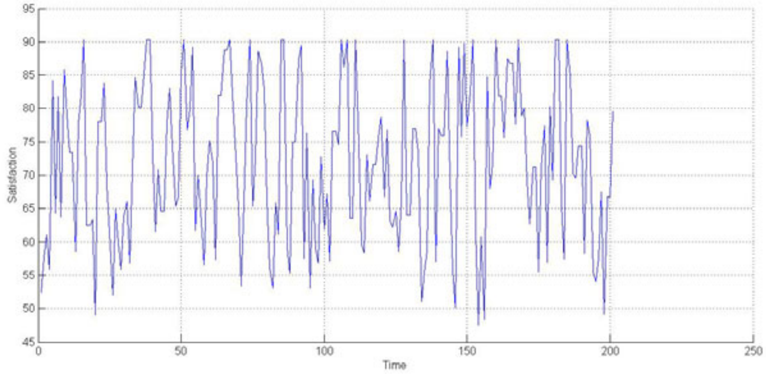

Fig. 5. Satisfaction degree

response-time of 3.67 seconds and availability of 68.6% result a satisfaction degree of 82.8. Figure 4 shows a surface map for the system and the dependency of the satisfaction degree on the response-time and availability.

We evaluate the effectiveness of the fuzzy approach with a non-fuzzy approach with respect to the stability of the system in terms of number of times a service needs to be replaced. The fuzzy approach performs the adaptation based on the QoS satisfaction. Only if the result of the satisfaction is lower than a threshold a service replacement occurs. While in the non-fuzzy approach, the replacement decision is done based on the precise evaluation of the QoS value ranges. We have conducted our experiment 200 times, each time providing random data for the input parameters. Figure 5 illustrates the output (satisfaction degree) of the experiment. The satisfaction threshold was set to 70%.

Figure 6 represents the stability of the fuzzy and non-fuzzy systems. As it is shown, the number of service replacement in a non-fuzzy approach is much

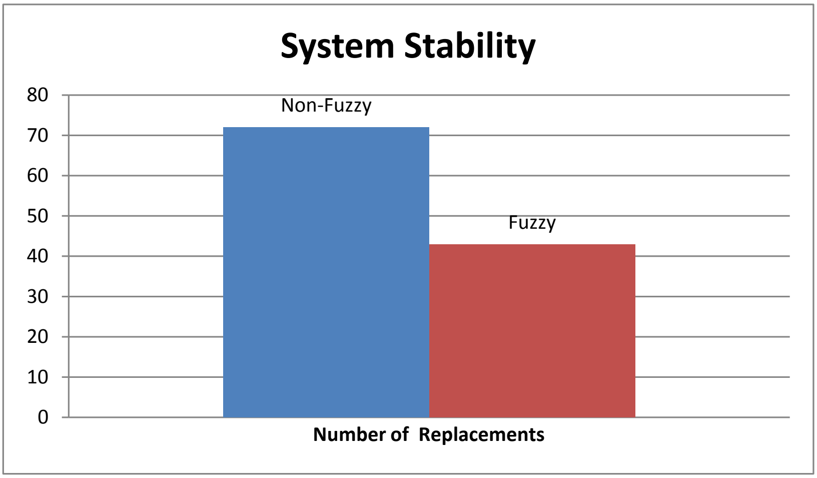

Fig. 6. System stability of using fuzzy and non-fuzzy approach

higher than when we apply a fuzzy approach. This actually is a direct proof of our approach. Using fuzzy parameters we allow partial satisfaction of the parameters. Therefore, the decision making for adaptation is not based on the precise evaluation of the quality ranges and it is rather imprecise and allows the parameters to be relaxed. The non-fuzzy approach involved the maximum number of service replacement which includes more queries for the service selection. This can results in a huge loss of time and money. The cost of establishing a new contract is also considerable.

7 Conclusions and Future Work

In this paper, we used fuzzy parameters for the QoS property descriptions of Web Services and a fuzzy approach is taken in order to select adaptation strategy. However, interpreting and presenting adaptation decisions based on fuzzy logic is still a hot research area that requires to be investigated more in the research community of software and service engineering.

In particular, we used linear *trapezoidal* membership function for the sake of simplicity. Currently, we are conducting more experiment to investigate the usage of *Gaussian* distribution function and *Sigmoid* curve that have the advantages of being smooth and non-zero all the time.

As for the future work, we aim to continue exploring the use of fuzzy parameters for the QoS matching. Applying more sophisticated functions using AI to Map the satisfaction degree to the appropriate adaptation decision might be worth exploring. However, there are still challenges that need to be addressed. For example, to what extent a parameter could be relaxed yet consider no violation? We also plan to incorporate more QoS parameters for calculating the overall satisfaction degree that influence the process of decision making. This requires the definition of more complex rules that represent the relation and dependencies between parameters.

Last but not least, applying an appropriate decision making requires an analytical evaluation based on a cost model. We would like to know under which circumstances the proposed approach is beneficial considering both QoS and business value criteria.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

References

1. Di Nitto, E., Kazhamiakin, R., Mazza, V., Bucchiarone, A., Cappiello, C., Pistore, M.: Design for adaptation of service-based applications: Main issues and requirements. In: The Fifth International Workshop on Engineering Service-Oriented Applications: Supporting Software Service Development Lifecycles, WESOA (2009)
2. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM 26(11), 832–843 (1983)
3. Andrikopoulos, V., Fugini, M., Papazoglou, M.P., Parkin, M., Pernici, B., Siadat, S.H.: Qos contract formation and evolution. In: EC-Web, pp. 119–130 (2010)
4. Brown, G., Cheng, B.H.C., Goldsby, H., Zhang, J.: Goal-oriented specification of adaptation requirements engineering in adaptive systems. In: SEAMS 2006: Proceedings of the 2006 International Workshop on Self-Adaptation and Self-Managing Systems, pp. 23–29. ACM, New York (2006)
5. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: Qos-aware replanning of composite web services. In: ICWS, pp. 121–129 (2005)
6. Cappiello, C., Pernici, B.: Quality-aware design of repairable processes. In: The 13th International Conference on Information Quality (ICIQ 2008), pp. 382–396 (2008)
7. Chafle, G., Dasgupta, K., Kumar, A., Mittal, S., Srivastava, B.: Adaptation in web service composition and execution. In: ICWS, pp. 549–557 (2006)
8. Chafle, G., Doshi, P., Harney, J., Mittal, S., Srivastava, B.: Improved adaptation of web service compositions using value of changed information. In: ICWS, pp. 784–791 (2007)
9. Cheng, B.H., Sawyer, P., Bencomo, N., Whittle, J.: A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 468–483. Springer, Heidelberg (2009)
10. Cheung, R., Cao, J., Yao, G., Chan, A.T.S.: A fuzzy-based service adaptation middleware for context-aware computing. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.-H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 580–590. Springer, Heidelberg (2006)
11. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program 20, 3–50 (1993)

12. Cock, M.D., Chung, S., Hafeez, O.: Selection of web services with imprecise QoS constraints. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007, pp. 535–541. IEEE Computer Society, Washington, DC, USA (2007)
13. Fenton, N., Neil, M.: Making decisions: using bayesian nets and mcda. Knowledge-Based Systems 14(7), 307–325 (2001)
14. Halpern, J.Y.: Reasoning about Uncertainty. MIT Press, Cambridge (2003)
15. Harney, J., Doshi, P.: Adaptive web processes using value of changed information. In: International Conference on Service-Oriented Computing (ICSOC), pp. 179–190 (2006)
16. Kritikos, K., Plexousakis, D.: Semantic QoS-based web service discovery algorithms. In: ECOWS, pp. 181–190 (2007)
17. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic symbolic model checking with PRISM: A hybrid approach. In: Katoen, J.-P., Stevens, P. (eds.) TACAS 2002. LNCS, vol. 2280, pp. 52–66. Springer, Heidelberg (2002)
18. Letier, E., van Lamsweerde, A.: Reasoning about partial goal satisfaction for requirements and design engineering. SIGSOFT Softw. Eng. Notes 29, 53–62 (2004)
19. Li, P., Comerio, M., Maurino, A., De Paoli, F.: Advanced non-functional property evaluation of web services. In: Proceedings of the 2009 Seventh IEEE European Conference on Web Services, ECOWS 2009, pp. 27–36. IEEE Computer Society, Washington, DC, USA (2009)
20. Li, P., Comerio, M., Maurino, A., De Paoli, F.: An approach to non-functional property evaluation of web services. In: Proceedings of the 2009 IEEE International Conference on Web Services, ICWS 2009, pp. 1004–1005. IEEE Computer Society, Washington, DC, USA (2009)
21. Martín-Díaz, O., Cortés, A.R., Benavides, D., Durán, A., Toro, M.: A quality-aware approach to web services procurement. In: Benatallah, B., Shan, M.-C. (eds.) TES 2003. LNCS, vol. 2819, pp. 42–53. Springer, Heidelberg (2003)
22. Martín-Díaz, O., Cortés, A.R., García, J.M., Toro, M.: Dealing with fixable and non-fixable properties in service matchmaking. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 228–237. Springer, Heidelberg (2010)
23. De Paoli, F., Palmonari, M., Comerio, M., Maurino, A.: A meta-model for non-functional property descriptions of web services. In: Proceedings of the 2008 IEEE International Conference on Web Services, pp. 393–400. IEEE Computer Society, Washington, DC, USA (2008)
24. Pernici, B., Siadat, S.H.: Adaptation of web services based on QoS satisfaction. In: WESOA 2010: Proceedings of the 6th International Workshop on Engineering Service-Oriented Applications. Springer, Heidelberg (2010)
25. Schäfer, M., Dolog, P., Nejdl, W.: An environment for flexible advanced compensations of web service transactions. ACM Trans. Web 2, 14:1–14:36 (2008)
26. Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., Bruel, J.-M.: Relax: Incorporating uncertainty into the specification of self-adaptive systems. In: RE, pp. 79–88 (2009)
27. Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., Bruel, J.-M.: Relax: a language to address uncertainty in self-adaptive systems requirement. Requir. Eng. 15(2), 177–196 (2010)

28. Yen, J., Tiao, W.A.: A systematic tradeoff analysis for conflicting imprecise requirements. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, RE 1997, pp. 87–96. IEEE Computer Society, Washington, DC, USA (1997)
29. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
30. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: Qos-aware middleware for web services composition. IEEE Trans. Softw. Eng. 30(5), 311–327 (2004)
31. Zhang, J., Cheng, B.H.C.: Using temporal logic to specify adaptive program semantics. Journal of Systems and Software 79(10), 1361–1369 (2006)