Towards an ideal service QoS in fuzzy logic-based adaptation planning middleware

Mounir Beggas, Lionel Médini, Frédérique Laforest, Mohamed Tayeb Laskri

▶ **To cite this version:**

Mounir Beggas, Lionel Médini, Frédérique Laforest, Mohamed Tayeb Laskri. Towards an ideal service QoS in fuzzy logic-based adaptation planning middleware. Journal of Systems and Software, 2014, 92, pp.71-81. ⟨10.1016/j.jss.2013.07.023⟩. ⟨hal-01010164⟩

HAL Id: hal-01010164

https://hal.science/hal-01010164v1

Submitted on 19 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Elsevier Editorial System(tm) for Journal of Systems and Software
Manuscript Draft

Manuscript Number:

Title: Towards an Ideal Service QoS in Fuzzy Logic-based Adaptation Planning Middleware

Article Type: Special Issue: Middleware Mobile

Keywords: mobile middleware, adaptation planning, context aware computing, fuzzy logic, QoS

Corresponding Author: Mr Mounir Beggas,

Corresponding Author's Institution:

First Author: Mounir Beggas

Order of Authors: Mounir Beggas; Lionel Médini, Associate Professor; Frederique Laforest, Professor; Mohamed Tayeb Laskri, Professor

Abstract: Mobile applications require an adaptation phase so as to adapt to the user context. Utility functions or rules are most often used to make the adaptation planning or decision (i.e. select the best adapted variant for each required service). Fuzzy controllers are used when it is difficult or even impossible to construct precise mathematical models. In the case of mobile applications, the large number of Quality of Service (QoS) and context parameters causes rule number explosion. An exponential increase in the number of rules increases the processing time of the adaptation planning. To reduce the processing time and simplify the fuzzy control system, we propose the concept of ideal QoS model. With it, fuzzy values of ideal QoS parameters are calculated using a fuzzy control system to fit the context state and user preferences. A fuzzy logic similarity method based on fuzzy sets and fuzzy operators is proposed to select among all service variants, the variant having the nearest QoS values to the ideal. Experiments show that our approach can significantly improve both the processing time and the number of rules when selecting the variant that well adapts to environment changes.

Cover Letter

Corresponding author : Mounir Beggas

Emails: mbeggas@gmail.com, mounir.beggas@liris.cnrs.fr

Address: Batiment Nautibus
43, boulevard du 11 novembre 1918
69622 Villeurbanne cedex, France.

Tel. +334 27 46 57 49

Fax. +334 72 43 15 36

Note: I submitted the same paper with the same list of authors but with another title. Old version is registered with:

Ref. No.: JSS-D-12-00774
Title: Augmenting Mobile Middleware with a Fuzzy Logic Based Adaptation Planning
Journal of Systems and Software

The new title is: “Towards an Ideal Service QoS in Fuzzy Logic-based Adaptation Planning Middleware”.

Please ignore the old version of the paper when reviewing.

***Highlights (for review)**

- We propose an approach based on fuzzy logic for adaptation planning middleware
- Addressed problem: the explosion in the number of rules that increases execution time
- We separately calculate each QoS value of an ideal variant for a given context state
- We define a fuzzy similarity method to select the most similar variant to the ideal
- Experiments show our approach both reduces the number of rules and execution time

*Manuscript
Click here to view linked References

Towards an Ideal Service QoS in Fuzzy Logic-based Adaptation Planning Middleware

Mounir Beggas[1,d], Lionel Médini[b], Frederique Laforest[c], Mohamed Tayeb Laskri[d]

[a] *Université de Lyon, CNRS, INSA-Lyon - LIRIS UMR5205*
[b] *Université de Lyon, , CNRS, Université Lyon 1- LIRIS UMR5205*
[c] *Université de Lyon, LT2C,Telecom Saint Etienne*
[d] *Department of computer science, University of Annaba, Algeria*

Abstract

Mobile applications require an adaptation phase so as to adapt to the user context. Utility functions or rules are most often used to make the adaptation planning or decision (i.e. select the best adapted variant for each required service). Fuzzy controllers are used when it is difficult or even impossible to construct precise mathematical models. In the case of mobile applications, the large number of Quality of Service (QoS) and context parameters causes rule number explosion. An exponential increase in the number of rules increases the processing time of the adaptation planning. To reduce the processing time and simplify the fuzzy control system, we propose the concept of *ideal QoS model*. With it, fuzzy values of ideal QoS parameters are calculated using a fuzzy control system to fit the context state and user preferences. A fuzzy logic similarity method based on fuzzy sets and fuzzy operators is proposed to select among all service variants, the variant having the nearest QoS values to the ideal. Experiments show that our approach can significantly improve both the processing time and the number of rules when selecting the variant that well adapts to environment changes.

Keywords: mobile middleware, adaptation planning, context aware computing, fuzzy logic, QoS.

Email addresses: mounir.beggas@liris.cnrs.fr (Mounir Beggas), lionel.medini@univ-lyon1.fr (Lionel Médini), frederique.laforest@telecom-st-etienne.fr (Frederique Laforest), laskri@univ-annaba.org (Mohamed Tayeb Laskri)

1. Introduction

Adaptivity is required to overcome the variability of mobile context-aware environments. In these environments systems are designed with adaptive properties that make them reconfigurable for different situations. A service-based application is modeled as a composition of service components. A service component can be delivered by different service variants. The main characteristic of service variants is that they maintain the functional properties of the service component, while varying its extra-functional characteristics. The purpose of context-aware self-adaptive systems can be considered as the adjustment of its extra-functional properties to the perceived context [1]. In this purpose, adaptation planning is the process that deduces an adaptation plan to adjust extra-functional properties to fit context conditions.

Mobile context-aware environments involve a large number of parameters: system resources, environment perceived parameters and user preferences. In this case, formulating an analytical model to deal with adaptation decision is a complex task [2, 3, 4] and it has always been a source of confusion for the developers.

Fuzzy control [5, 6] is a control technique that uses fuzzy sets and human-like fuzzy rules [7] to define nonlinear systems. It has been successfully applied to various application-specific network QoS management systems [8, 9], service QoS management systems [2, 10] and adaptive resource management for enterprise services [11, 12]. Fuzzy logic can be used for adaptation planning design, allowing context and application QoS parameters to be expressed with linguistic variables and membership functions. However, the large number of QoS and context parameters in a mobile operating environment causes the rule explosion problem, in which an exponential increase in the number of rules increases the processing time. For example, in the case of a service being aware of 4 context parameters and having 5 QoS parameters , and each context and QoS parameter having 3 linguistic values, $4^3 \times 5^3 = 8000$ fuzzy rules are needed for adaptation decision.

Contribution. In this paper we propose an adaptation planning approach based on fuzzy control techniques for mobile middleware. We address the problem of the explosion in the number of rules by proposing the concepts of *ideal QoS* and *ideal variant*. Ideal QoS represents the QoS parameter value that best fits the current context state. The set of ideal QoS parameter

values form the ideal variant. To select the best service variant, we calculate the degree of similarity of their QoS parameters to the set of ideal QoS. For that, a fuzzy logic similarity method is proposed. The variant having the nearest QoS values to the ideal (ie. the highest degree of similarity) is the best one.

Given a large number of QoS parameters and context parameters the proposed adaptation planning strategy can reduce the total number of fuzzy rules and the decision processing time. That can be of importance especially if the decision process is carried out on a mobile device. The proposed approach does not focus on quality of results, it ensures the same quality of results as in classical fuzzy based approaches with a reduction in the needed processing time. For the service of the previous example, adaptation planning is performed using 5 fuzzy control systems, one for each ideal QoS parameter, with $4^3 = 64$ fuzzy rules. The total number of rules is $64 \times 5 = 320$ fuzzy rules, which is smaller than the 8000 previously needed.

Paper organization. The remainder of this paper is structured as follows. Section 2 presents a background on a fuzzy logic and fuzzy sets. Section 3 describes the related work. A reference example is presented in section 4. In section 5 we present a definition and a modeling of context, as well as QoS parameters where we introduce the concepts of QoS fuzzy parameters. In Section 6, we present the proposed approach for adaptation planning based on fuzzy logic. Section 7 provides experimental results and evaluates the efficiency of the proposed approach. Section 8 concludes the paper and opens perspectives for the future work.

2. Background

Fuzzy control is used in control problems for which it is difficult to construct precise mathematical models [5]. Using a fuzzy control system is a convenient way to map the input space to the output space using intuitive if-then rules. These rules are based on the knowledge of an expert. For all input variables, crisp values are converted into appropriate linguistic variables using fuzzy sets [13].

Fuzzy sets are represented by linguistic terms that build linguistic variables. The linguistic variables have their possible states defined in a universe of discourse U, represented by these linguistic terms. A fuzzy set A, in a universe of discourse U, is characterized by a membership function

$\mu_A : U \rightarrow [0, 1]$, where $\mu_A(x)$ indicates the membership degree of the crisp values x to the fuzzy set A [14]. The representation of fuzzy sets can be done in many ways as, for example, triangular, trapezoidal or Gaussian. An example of trapezoidal fuzzy sets is shown in Figure 1.

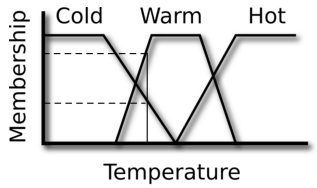


Figure 1: Example of trapezoidal fuzzy sets [15]

In figure 1, *Cold, Warm* and *Hot* are linguistic terms associated with fuzzy sets. These fuzzy sets are defined by the corresponding membership functions. Each crisp value matches each term to some degree in the interval [0, 1], so, for example, a membership function might define $10°C$ as (0.8 cold, 0.4 warm, 0.0 hot).

Once linguistic variables and membership functions are defined, a fuzzy control system processes the data using linguistic fuzzy rules. There are different kinds of linguistic fuzzy rules proposed in the literature [16]. The most used rule structure considers a linguistic variable in the rule consequent as follows:

$$\textbf{If } (x_1 \textit{ is } A_1) \textit{ and } \ldots \textit{ and } (x_n \textit{ is } A_n) \textbf{ then } y \textit{ is } B,$$

with x_i and y respectively being the input and output linguistic variables, and A_i and B being the linguistic terms.

The fuzzy proposition (x_i *is* A_i) is evaluated using the corresponding membership function as $\mu_{A_i}(x_i)$. The conjunction operator *and* and the disjunction operator *or* can be used in the rule antecedent. The *and* operator is evaluated by *min* or *prod* functions, and the *or* operator is evaluated by *max* or *probOR* (probabilistic OR, also known as the algebraic sum) functions.

The fuzzy inference process, as shown in Figure 2, consists of three stages: fuzzification, reasoning by inference engine, and defuzzification. During fuzzification, predefined membership functions for each linguistic variable are used to determine the degree of membership of a crisp value. Then, the inference engine refers to a predefined fuzzy rule base to derive the degree of membership of output linguistic variables. Once the output linguistic values are

available, in defuzzification, the final crisp values are produced from the output linguistic values.

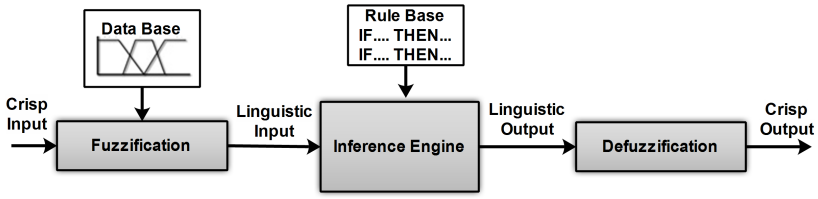


Figure 2: Fuzzy Control Process

3. Related Work

Context-aware adaptation refers to the ability of computing systems to adapt their behavior or structure to highly dynamic environments without explicit intervention from users. Adaptation is handled by an adaptation control loop [17] that (1) senses the relevant context changes, including variations in computational resources, operating environment contexts and user needs, (2) makes decision on the adaptation by planning the suitable reconfiguration, (3) and finally executes these decisions by dynamically reconfiguring the system.

In Service Based Applications (SBA), services are generic reusable and composable entities. Services are implemented by service variants. Each service variant performs the same general task and shares the same functional parameters. Service variants differ by their extra-functional parameters, aka QoS parameters. Application reconguration using SBAs consists in selecting the most adapted service variant for each generic service, in a given context [18].

In context-aware SBAs, the adaptation control loop is controlled by a middleware layer [2, 19, 20]. Figure 3 illustrates the main components of this middleware. The context manager monitors and analyzes the environment context; on context changes, an adaptation planning process is started. The adaptation planning deduces the suited adaptation plan for the current context situation. The reconfiguration engine reconfigures the application services by executing the adaptation plan. The reconfigured service based application is composed of selected service variants and the necessary bindings between them.

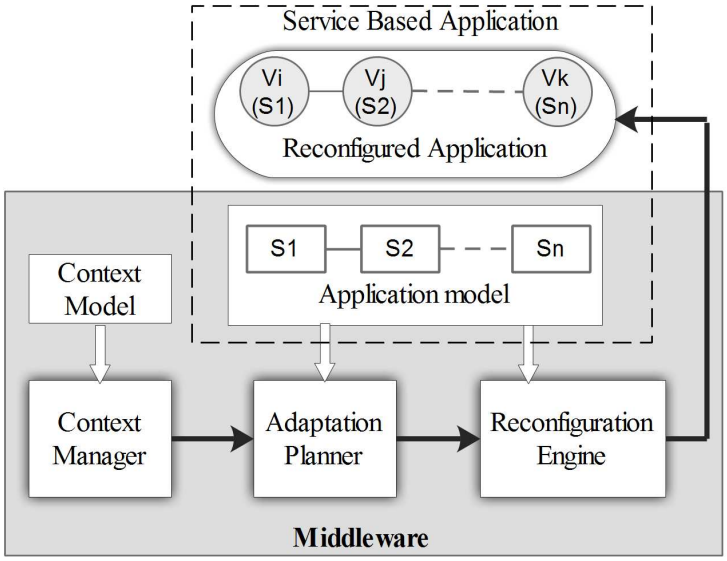


Figure 3: Middleware Adaptation planning for Service Based Applications

3.1. Adaptation Planning Strategies

Researchers have proposed several techniques to simplify SBA reconfiguration. In [21, 22], the adopted adaptation strategy is based on situation-action rules. Each rule is formulated in terms of conditions and actions. An obvious limitation of rule-based approaches is the imposed binary decision logic since each condition is either evaluated to true or false. Another limitation of rule-based approaches for SBA adaptation is that they do not allow dynamically adding new variants without redefining the whole set of rules. Furthermore, application developers need to explicitly describe, at design time, the adaptive reaction of the system in response to any relevant context change.

To overcome these limitations, Floch et al [4] use a utility function for adaptation decision. The utility function is mainly used in some micro-economics decision problems [23]. It is also applied in service selection to satisfy constraints and user preferences [24]. Utility functions are mathematical artifacts that map each possible system state, or alternative implementations, to a real scalar value. For SBA reconfiguration, a distinct utility function is defined for each service variant. The highest utility score indicates the most suitable variant for the current context. In MADAM and MUSIC [4, 19], the middleware uses utility functions to calculate utility scores for

each application variant. Defining utility functions, especially for complex applications with multiple QoS and context parameters is a difficult task [2, 3, 4]. Utility functions are based on variant properties prediction, current context situation and user preferences. Therefore, application developers need to specify application variants in details and formulate every possible adaptation aspect in mathematical equations.

3.2. Fuzzy Logic Based Adaptation Planning Strategies

Fuzzy logic based adaptation reasoning can be considered as an extension of the work on utility functions using a fuzzy control system. Pernici et al. [10] propose a fuzzy approach to support service adaptation and evolution. For this purpose, they define fuzzy parameters for QoS using membership functions, and use a fuzzy control system to deal with adaptation decision. Gmach et al [12] use a fuzzy logic controller for the adaptive management of resource allocation for enterprise services to guarantee negotiated service level agreements. Acampora et al [25] use a fuzzy logic controller to deduce the appropriate actions according to the environment context, for ambient intelligence applications.

The large number of QoS and context parameters causes the rule explosion problem. An increase in the number of rules increases the processing time. Hierarchical fuzzy systems [26] were introduced to reduce the number of rules using hierarchical fuzzy control. In such systems, correlated linguistic variables are hierarchically inferred and grouped into abstract linguistic variables. Chuang et al [2] propose a mobile service QoS management system based on fuzzy control system and the fuzzy decision function proposed in [27]. Each mobile service is described by a satisfactory user, environment and resource QoS factors that will be affected if the service is selected. The fuzzy control system deduces the adaptation importance factors that will be used by the fuzzy decision function to select the appropriate service. They use a hierarchical fuzzy control system to reduce the number of rules by hierarchically grouping correlated QoS linguistic variables into abstract linguistic variables. Nevertheless, the rule reduction level depends on the correlation between linguistic variables.

In [28, 29], the proposed fuzzy adaptation model measures experimentally the most suitable values (best value) of context for each service variant. To select the best variant they use the so-called fitness function. This function uses fuzzy set theory to calculate the fuzzy distance between the context situation and the calculated best value of context. The fitness function equals

the inverse of sum of distances. The variant with the largest fitness degree is selected. Using this adaptation model, to calculate experimentally the best values is not easy especially if there is a large number of parameters.

In the discussed work, fuzzy logic control technique is used to define the decision function but with a limited number of parameters. A large number of parameters causes the rule explosion problem that has not been significantly treated. In our work, ideal QoS and ideal variant concepts are introduced to reduce the number of rules and reduce the processing time. Ideal QoS values are calculated for each context state using fuzzy control systems. To select the most satisfactory variant, we propose a fuzzy similarity method.

4. Reference Example

We consider an instant video messaging application. The application is composed of two services CamVideo and GUI. The CamVideo service captures and sends image sequences. The GUI service is a graphical user interface that displays the images received from another application. CamVideo has two QoS parameters: sequence rate (number of images/sec) and image quality (resolution in pixels). Using them, CamVideo variants are provided with different QoS levels. GUI is provided in a single variant. The application is designed to be aware of context changes. Context parameters are: bandwidth, battery level and video zone size.

To use a fuzzy logic technique, we define linguistic terms and membership functions, for each context and QoS parameter. Each parameter has three linguistic terms:

- image-quality (HighQuality, MediumQuality, LowQuality),
- sequence-rate (HighRate, AvgRate, LowRate),
- bandwidth (Good, Average, Poor),
- battery-level (Low, Medium, High) and
- video-zone-size (Large, Normal, Small).

5. Definitions and Application Parameters Modeling

The definition and formulation of concepts related to service adaptation planning and their fuzzy extension are introduced in this section.

5.1. Service Modeling

Definition 1 (Service). A service is an abstract component that defines a class of service variants offering similar functionalities. It comprises functional and extra-functional (QoS) parameters. Service parameters are defined as follows: $S = (F,\ QoS)$.

The functional interface $F(S) = (I,\ O)$ specifies the service inputs and outputs. Extra-functional parameters denoted $QoS(S)$ are detailed in section 5.2.

Definition 2 (Service variant). A service variant $V_i(S)$ or simply a variant is an instance of the service S. All variants of S share the same functional interface and are defined with specific values of QoS parameters.

$$V(S) = \{V_i(S),\ V_i(S) = (F(S),\ QoS(V_i(S))) \mid i = 1 \dots k\}, \tag{1}$$

with k being the number of variants of the service S. $QoS(V_i(S))$ is defined in the next section.

5.2. QoS Parameters Modeling

In this section, we present a formal definition of QoS parameters. It is an extension of the work in [28, 19, 10]. We extend this work by defining QoS fuzzy parameters for such service descriptions. QoS fuzzy parameters are represented by linguistic terms and their associated fuzzy sets.

QoS parameters of the service S are defined as follows:

$$QoS(S) = \left\{Q^i \mid i = 1 \dots n_Q\right\}, \tag{2}$$

where n_Q is the number of QoS parameters for the service S and Q^i is a class of QoS parameters defined on a domain $dom(Q^i)$.

The set of QoS parameter values of the i^{th} variant of the service S is represented as:

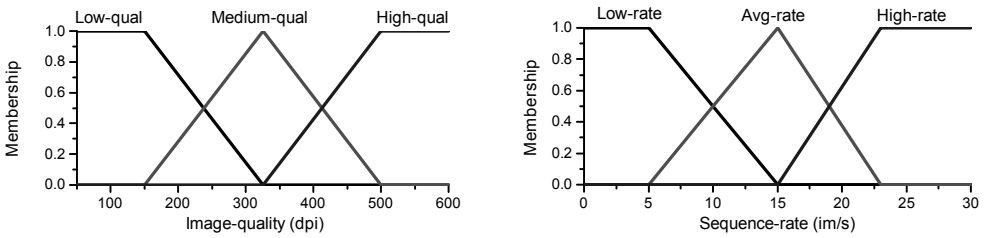
$$QoS(V_i(S)) = \left\{q_i^j \in dom(Q^j) \mid j = 1 \dots n_Q\right\}, \tag{3}$$

The fuzzy value of q_i^j is measured by the membership degrees to the fuzzy sets defined for the relevant QoS parameter class. So, the final fuzzy QoS parameters of a variant is defined as a matrix.

$$fQoS(V_i(S)) = \left\{\left(\mu_k^j(q_i^j),\ k = 1 \dots n_j\right) \mid \mu_k^j(q_i^j) \in [0,\ 1],\ j = 1 \dots n_Q\right\}, \tag{4}$$

where n_j is the number of membership functions μ_k^j defined for the QoS parameter Q^j.

Example. To illustrate the previous definitions with *CamVideo* service, we use membership functions for each linguistic term depicted in Figure 4.



(a) Image-quality (b) Sequence-rate

Figure 4: Linguistic terms and membership functions of application QoS parameters

QoS parameter values of $V_1(CamVideo)$ are:

$$QoS(V^1_{CamVideo}) = \{\big(8\, im/sec,\ 280\, dpi\}$$
$$fQoS(V^1_{CamVideo}) = \{(0.6,\ 0.4,\ 0.0),\ (0.2,\ 0.8,\ 0.0)\}$$

This variant has $8im/sec$ rate which corresponds to $\mu_{low_qual}(8) = 0.4$ low-rate, 0.6 avg-rate and 0.0 high-rate. The supported image quality is $280\, dpi$ which corresponds to 0.2 low-quality, 0.8 medium-quality and 0.0 high-quality.

5.3. Context Modeling

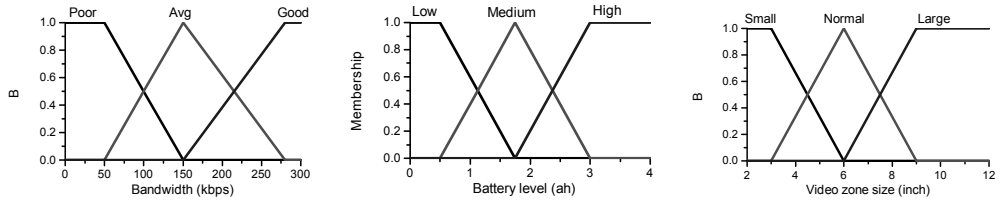
The context is modeled through a finite set of context parameters. In particular, for a given application, we define its contextual environment as a set of n_c context parameters $\{C_i \mid i = 1 \ldots n_c\}$.

Definition 3 (Context state). A context state corresponds to an assignment of values to context parameters at some point in time. In particular, a context state $W_t = \{c_i \mid i = 1 \ldots n_c\}$.

The fuzzy value of c_i is measured by the membership degrees to fuzzy sets defined for the relevant context parameter in the same way as in equation (4).

Example. The Linguistic terms and membership functions of context parameters of the example are illustrated in Figure 5. An example of context state can be:

$$W_t = \{75\, kbps,\ 2.5\, ah,\ 2.5\, in\}\,.$$



(a) Bandwidth (b) Battery level (c) Video zone size

Figure 5: Linguistic terms and membership functions of context parameters.

The fuzzy value of W_t is then:

$$fW_t = \{(0.70,\ 0.30,\ 0.0),\ (0.0,\ 0.4,\ 0.6),\ (1.0,\ 0.0,\ 0.0)\}\,.$$

5.4. Application Modeling

Definition 4 (Application). An application is a service based application modeled by a composition of services. Let the application be defined as:

$$A = \{S_i \mid i = 1 \ldots n_s\}\,,$$

with n_s being the number of application services.

The running application is performed by a dynamic binding between service variants. In a given context state W_t, the running application is:

$$A(W_t) = \{V_j(S_i) \in V(S_i) \mid S_i \in A\ and\ i = 1 \ldots n_s,\ j \in [1 \ldots k_i]\},$$

k_i being the number of variants of the service S_i.

Definition 5 (Adaptation planning). An adaptation planning is the decision making process that maps the current context state to a set of suitable service variants.

The adaptation decision can be considered as the process of applying on variants extra-functional parameters a function of the form:

$$f\,(W_t,\ QoS(V_i))\,.$$

This function calculates the satisfaction degree of the variant that has the QoS parameters $QoS(V_i)$ in the context state W_t. Therefore, the decision function determines for each context state the satisfaction degree of each variant. Figure 6 shows the form of a fuzzy control system for the adaptation decision function.

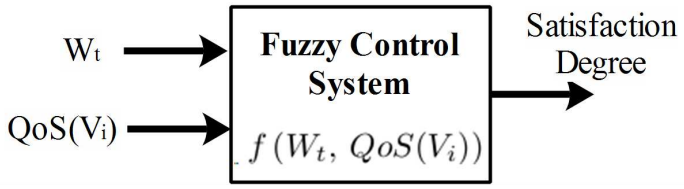


Figure 6: Fuzzy control system based adaptation decision functions

Example. In the reference example, the application is composed of two services

$$A = \{CamVideo,\ GUI\}.$$

QoS parameters for the service CamVideo are:

$$QoS(CamVideo) = \{sequence\text{-}rate,\ image\text{-}quality\}.$$

The running application in a given state W_t is:

$$A(W_t) = \{V_2(CamVideo), V_1(GUI)\}.$$

The total number of fuzzy rules forming the previous decision function using QoS and context parameters of the reference example is 210 rules. An example of these rules is shown in Figure 7.

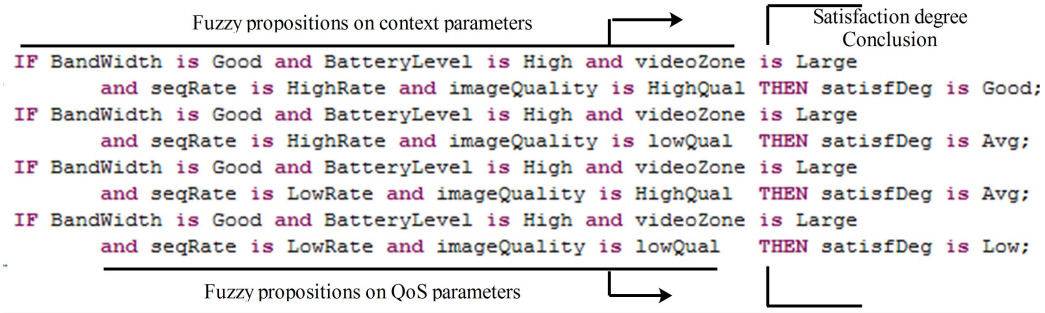


Figure 7: Example of fuzzy rules deducing the degree of satisfaction

The proposed approach of decision process based on fuzzy logic theory is presented in the next section.

6. Fuzzy Logic Based Adaptation Decision

In this paper, application management is based on IBMs MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) reference model for autonomic control loops [30]. Our work focus on the "Plan" step of this model. The proposed planning strategy is based on (i) finding the ideal variant for a given context state, by introducing the concept of Ideal QoS, and (ii) calculating the best variant, using a fuzzy similarity method.

The adaptation planning is split up into two parts as shown in Figure 8. The first part is based on a set of Ideal QoS fuzzy controllers. These controllers contain specific fuzzy rules used to infer Ideal QoS values. These values define the Ideal variant independently of existing variants. The second part calculates the fuzzy similarity between each variant and the ideal variant to allow the selection of the variant having the highest degree of similarity.

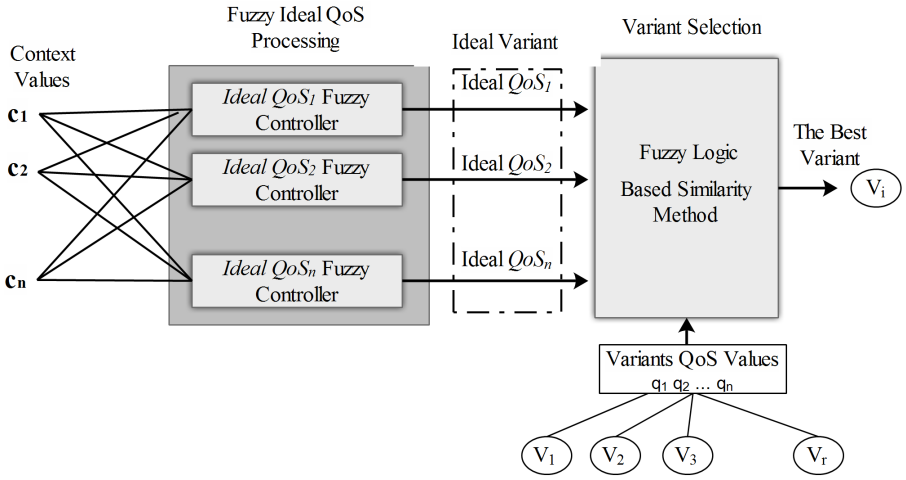


Figure 8: Fuzzy logic adaptation planning process

Our approach reduces the execution time by reducing the number of rules. Furthermore, the fuzzy inference process is applied only once whatever the number of variants.

6.1. Ideal QoS and Ideal variant

We herein propose the concepts of *Ideal QoS* and *Ideal variant*. Ideal QoS represents the value of a specific QoS parameter of a service definition

that best fit the current context state. The set of Ideal QoS parameter values define an Ideal variant that should be targeted for variant selection.

The ideal QoS values ($IQoS$) for the service S are modeled in the same way as variant qualities in equation (3) as follows:

$$IQoS(S) = \left\{ Iq^i \mid i = 1 \dots m \right\}. \tag{5}$$

The fuzzy ideal QoS values $fIQoS$ are measured by the membership degrees to fuzzy sets defined for the relevant QoS parameter in the same way as in equation (4).

Introducing this intermediary step between the context and the variant models aims at reducing the complexity of the mapping between these two models. Indeed, the mapping problem is divided into as many parts as the service interface defines QoS parameters. Our approach first addresses the rule explosion problem by reducing the number of rules and therefore the processing time. It also eases the rule design by simplifying the scope of these rules.

6.2. Ideal QoS fuzzy Rules and fuzzy controller

Each ideal QoS parameter value is deduced from context parameters using a separate fuzzy control system (Figure 8). In the fuzzy ideal QoS processing part, fuzzy rules specify the expected value of each QoS parameter according to values of context parameters.

In classical fuzzy control decision functions (Figure 7), we observe two kinds of antecedent fuzzy propositions in the same rule; one kind concerns context parameters, for instance *(BandWidth is Good and batteryLevel is High and videoZone is Large)*. The other kind concerns QoS parameters, for example *(seqrate is LowRate and imageQuality is LowQuality)*. To specify the satisfaction degree, the rule designer makes a comparison between context and QoS propositions to deduce the suitable satisfaction degree. With a large number of rules, this comparison is not easy.

Instead of comparing between QoS and context propositions to deduce the satisfaction degree, our rules deduce the most satisfactory QoS propositions for a given context state. Rule antecedents are only composed of context parameters. Rule conclusions form fuzzy Ideal QoS values, which are used to calculate the crisp ideal QoS values. With this approach, only one QoS parameter type is concerned by a given rule and the concept of satisfaction degree is no longer needed. The satisfaction degree is replaced by the similarity degree (see section 6.3).

For each service QoS parameter, an Ideal QoS fuzzy controller is defined by the group of rules related to this QoS parameter (see Figure 8). In the worst case, all context parameters are inputs of this controller. Nevertheless, only context parameters correlated with this QoS parameter need to be taken into account for expressing the rules. This reduces even more the number of rules.

Figure 9 shows the fuzzy rules of the two fuzzy controllers that deduce the ideal QoS values for our example. To deduce the ideal *sequence-rate* (resp. *image quality*), the conditions only refers to the *bandwidth* and *battery-level* (resp. *bandwidth* and *video-zone*) context parameters. To deduce ideal QoS crisp values a fuzzy inference process is applied on each set of rules (cf. Section 2)

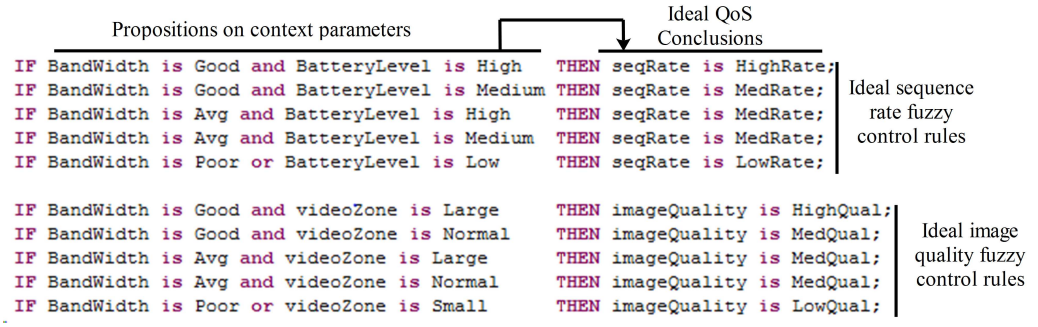


Figure 9: Fuzzy rules that deduce ideal QoS

To express the rule number reduction, we consider that each context and QoS fuzzy parameters have the same number of linguistic terms n_l. Let n_c be the number of context parameters and n_Q be the number of QoS parameters. In the worst case:

- the classical approach needs $n_l^{n_c+n_Q}$ rules, and
- our approach needs $n_l^{n_c} \times n_Q$ rules.

In our reference example, the rule base for a classical fuzzy control system would contain 210 rules as described in figure 7. Using our approach, only 10 rules are needed to deduce ideal QoS values.

Example. In a given context state, fuzzy controllers deduce the following ideal QoS values: 7 im/s for *sequence-rate* and 300 *dpi* for *image-quality*.

The fuzzy ideal QoS values are represented as follows:

$$fIQoS(CamVideo) = \{(0.7,\ 0.3,\ 0.0),\ (0.15,\ 0.85,\ 0.0)\}\,.$$

6.3. Variant Selection Using Fuzzy Similarity Method

The selected variant is the one that has the most similar QoS values to the calculated ideal QoS. We introduce a fuzzy similarity method to calculate the degree of similarity between the ideal QoS values of the ideal variant and QoS values of different candidate variants. The proposed fuzzy similarity method calculates the similarity degree between linguistic variables of ideal QoS and variants' QoS using fuzzy logic propositions and operators.

The similarity between the variant QoS values $fQoS(V_i)$ and ideal QoS values $fIQoS(S)$ is a value in [0, 1] that transcribes the fuzzy degree of similarity. It is calculated using fuzzy propositions by making a comparison between linguistic variables of all QoS parameters. Given that each membership function μ_j^k defines a linguistic term L_j^k; the similarity is calculated by the following equation:

$$Sim\left(fQoS(V_i),\ fIQoS\right) = \bigwedge_{j=1}^{n_Q} \left[\bigvee_{k=1}^{n_j} \left(q_i^j \ \textbf{\textit{is}}\ L_j^k \ \wedge\ Iq^j \ \textbf{\textit{is}}\ L_j^k \right) \right], \qquad (6)$$

where n_Q is the number of QoS parameters and n_j is the number of linguistic terms (membership functions) of the QoS parameter Q^j. The fuzzy proposition $(q_i^j \ \textbf{\textit{is}}\ L_j^k)$ represents the degree of membership of the current QoS value to the linguistic term. It is calculated using the relevant membership function as follows: $\mu_j^k\left(q_i^j\right)$.

Using membership functions to evaluate fuzzy propositions, the equation (6) can be written in the following form:

$$Sim\left(fQoS(V_i),\ fIQoS\right) = \bigwedge_{j=1}^{n_Q} \left[\bigvee_{k=1}^{n_j} \left(\mu_j^k\left(q_j^k\right) \ \wedge\ \mu_j^k\left(Iq^k\right) \right) \right]. \qquad (7)$$

In fuzzy logic, the conjunction operator $\wedge$ can be calculated with *min* or *prod* functions and the disjunction operator $\vee$ can be calculated with *max* or *probOR* functions. For the evaluation of the similarity degree, The functions selected to represent those operators should be the same functions used by the inference engine in the Ideal QoS Fuzzy Control System (cf. section 6.2).

Example. We consider two variants of the CamVideo service: $V_1(CamVideo)$ and $V_2(CamVideo)$ with the QoS values depicted in Table 1. Their fuzzy ideal QoS values and those of ideal variant are depicted in Table 1.

Ideal and candidate Variants	Sequence rate)		Image quality	
	Crisp	Fuzzy	Crisp	Fuzzy
$Ideal(CamVideo)$	7	(0.7, 0.3, 0.0)	300	(0.1, 0.9, 0.0)
$V_1(CamVideo)$	6	(0.8, 0.2, 0.0)	280	(0.2, 0.8, 0.0)
$V_2(CamVideo)$	10	(0.5, 0.5, 0.0)	325	(0.0, 1.0, 0.0)

Table 1: Ideal and candidate variants QoS values

The similarity degree between the ideal QoS values of the ideal variant and QoS values of the variant $V_1(CamVideo)$ is calculated by the following formula:

$$\begin{gathered} Sim\left(QoS(V_1(CamVideo)),\ IQoS(CamVideo)\right) = \\ \Big[\ (\mu_{low\text{-}rate}(6) \wedge \mu_{lowrate}(7)) \vee (\mu_{med\text{-}rate}(6) \wedge \mu_{med\text{-}rate}(7)) \vee \\ (\mu_{high\text{-}rate}(6) \wedge \mu_{high\text{-}rate}(7))\ \Big] \bigwedge \Big[\ (\mu_{low\text{-}qual}(280) \wedge \mu_{low\text{-}qual}(300)) \vee \\ (\mu_{med\text{-}qual}(280) \wedge \mu_{med\text{-}qual}(300)) \vee\ (\mu_{high\text{-}qual}(280) \wedge \mu_{high\text{-}qual}(300))\ \Big] \\ = 0.7 \end{gathered}$$

Using the min (resp. max) function for operator $\wedge$ (resp. $\vee$), yhe similarity degree is calculated as follows:

$$\begin{gathered} Sim\left(QoS(V_1(CamVideo)),\ IQoS(CamVideo)\right) = \\ \min\left[\max\left(\min(0.7,\ 0.8),\ \min(0.3,\ 0.2),\ \min(0.0,\ 0.0)\right),\right. \\ \left.\max\left(\min(0.1,\ 0.2),\ \min(0.9,\ 0.8),\ \min(0.0,\ 0.0)\right)\right] = 0.7 \end{gathered}$$

In the same way, the similarity degree between the ideal variant and $V_2(CamVideo)$ is obtained as follows:

$$\begin{gathered} Sim\left(QoS(V_2(CamVideo)),\ IQoS(CamVideo)\right) = \\ \min\left[\max\left(\min(0.7,\ 0.5),\ \min(0.3,\ 0.5),\ \min(0.0,\ 0.0)\right),\right. \\ \left.\max\left(\min(0.1,\ 0.0),\ \min(0.9,\ 1.8),\ \min(0.0,\ 0.0)\right)\right] = 0.5 \end{gathered}$$

So, The variant $V_1(CamVideo)$ has the highest similarity degree and will be selected as the best candidate for this context state.

7. Experiments and results

We have implemented and tested our proposed approach using the fuzzy control language and tool JFuzzyLogic[1]. It is a Java implementation of fuzzy control language specification IEC 61131 part-7[2]. In our experiments, we used Mamdani type [31] for rules and inference models. To validate the proposed fuzzy similarity approach, we tested both operators for the fuzzy $\wedge$: *min* and *prod* functions (*max* and *probOR* for $\vee$).

To evaluate our approach, we implemented three decision making processes for the reference example with three different manners; (1) the proposed Fuzzy Similarity To Ideal (FuSTI) approach , (2) fuzzy control system and (3) hierarchical fuzzy control system. The effectiveness of the proposed approach is evaluated by a comparison between the results of the three systems with respect to execution time and number of rules. The experiments have been conducted on a Pentium Core i5 with 6 GByte RAM.

In the following we evaluate the adaptation quality results by comparing the results of the three systems. After that we evaluate the performance of the proposed approach in terms of execution time and number of rules.

7.1. Comparison of the adaptation quality

Our objective is to decrease processing time without decreasing adaptation quality. In this first evaluation step, our aim is to ensure that our adaptation (i.e. variant selection) results are equivalent to the results of the other approaches. Therefore, we do not intend to evaluate the intrinsic quality of the variant selection results.

To present the comparison of the selected variant between the three methods, we conducted the experiment 50 times, in different context states. Each time, we have applied the three methods. The results are illustrated in Figure 10. This figure indicates in each context state the selected variant using the three methods. The superposition of symbols means that they have selected the same variant. The figure shows that the selected variant is the same for the three methods in most context states.

The selected variant is the one that has the highest similarity to the ideal variant for our proposed approach (FuSTI) and the highest satisfaction degree for the other approaches. Figure 11 shows the similarity degree and

[1] `http://jfuzzylogic.sourceforge.net`
[2] `http://www.iec.ch/`

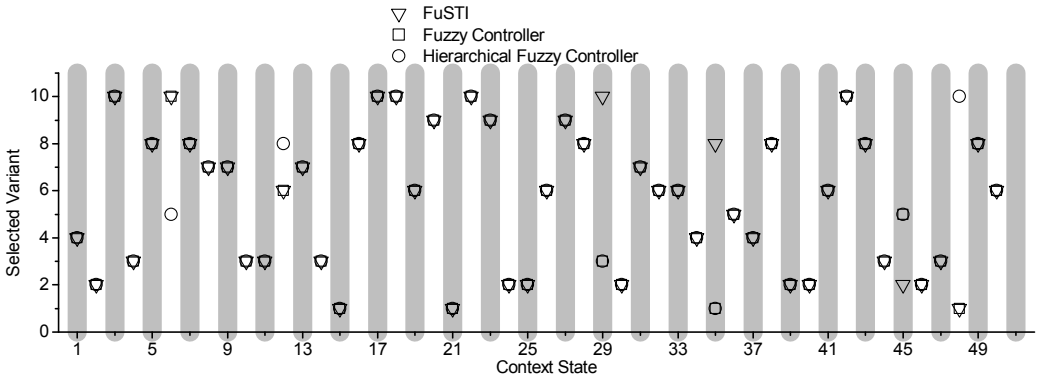


Figure 10: Selected variants in the three methods

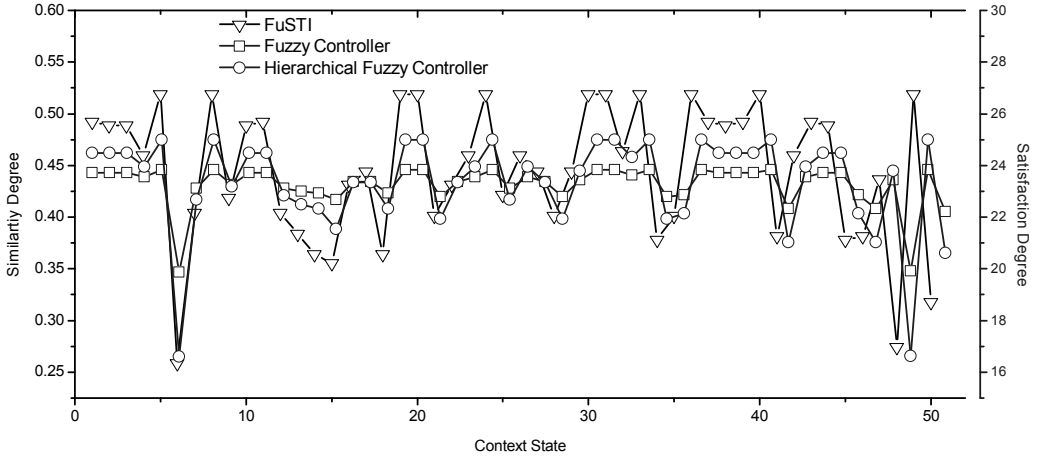


Figure 11: Selected variant similarity degree and satisfaction degrees of the three methods

the satisfaction degrees of the selected variants. We notice that they vary in the same direction: when the selected variant is very similar to the ideal in the proposed approach, it is more satisfactory for the other approaches and vice versa. That means that the compared approaches are not selecting just the same variant but with the same satisfaction level.

To compare the results of the three methods for each variant, we take values of fuzzy similarity and satisfaction degrees in different context states (states 1, 5, 6, 29, 48 and 50 of Figure 11). The results are presented in Figure 12. Figures 12(a), 12(b) and 12(f) illustrate the cases when the selected variant is the same in the three methods. The other figure results are taken

when the selected variant is not the same. The presented values show that the fuzzy similarities and the satisfaction degrees are always varied together in the same direction, even if they do not select exactly the same variant.

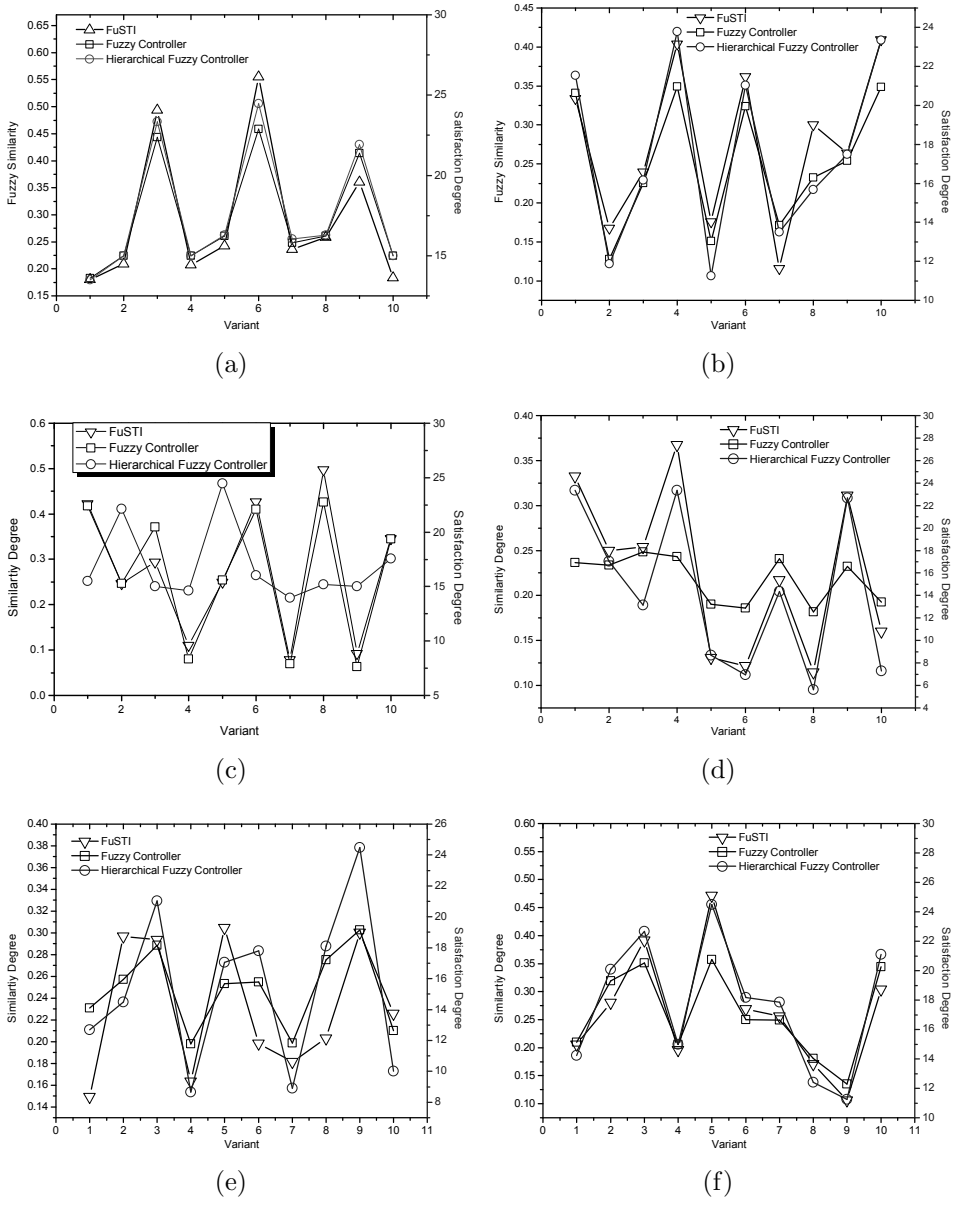


Figure 12: Similarity and satisfaction degrees for each variant in several context states

7.2. Performance Evaluation

To compare the number of fuzzy rules of the three approaches, we vary the number of QoS and context parameters from 3 to 8. This comparison is shown in Figure 13. To evaluate the effectiveness of the proposed approach we have varied the number of variants from 10 to 1500 and compare the execution time in the three approaches. Figure 14 illustrates the obtained results. Note that in this two figures, the vertical axis is plotted on a logarithmic scale.

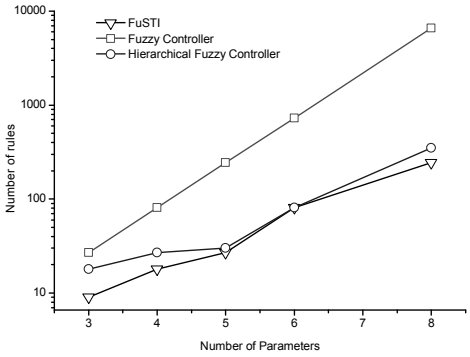


Figure 13: Number of rules as a function of the number of parameters

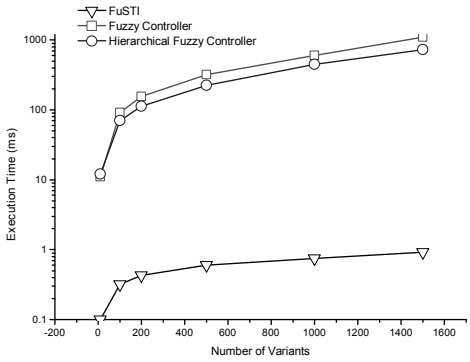


Figure 14: Execution time according to the number of variants

As shown in Figure 13, the number of rules increases linearly as a function of the number of parameters when applying the proposed approach. Although, it increases exponentially when applying the fuzzy control system. We notice that in hierarchical fuzzy control, the number of rules increases linearly with the number of parameters, but it depends on the correlation between parameters. In our proposition we can use a hierarchical fuzzy control to decrease even more the number of rules if the number of context parameters is large.

Figure 14 and Table 2 show that the proposed method reduces the execution time comparing to the other methods. With the proposed approach (FuSTI), the execution time varies linearly with the variation in the number of variant, but it is slowly increasing (it is always lower than $1 ms$). With the two other approaches, the execution time increases quickly with the number of variants. Using the proposed approach, adding to reduction in the number of rules, the fuzzy controller is only executed one time whatever the number of variants. Whereas, in the case of fuzzy controller and hierarchical fuzzy controller, the inference process is executed once for each variant.

Number of Variants	Execution Time (ms)		
	FuSTI	Fuzzy Cont	Hier Fuzzy Cont
10	0.11	11	11
100	0.32	92	70
200	0.43	156	113
500	0.61	321	225
1000	0.75	600	451
1500	0.92	1104	726

Table 2: Execution time using the three methods

The proposed approach achieves almost the same results in terms of selected variant, whilst ensuring a good processing time with a limited number of rules.

8. Conclusion and Future Work

In this paper, we propose an adaptation decision approach that aims at reducing the processing time during the planning step of a middleware adaptation for context-aware, service-based applications. This approach is based on the fuzzy logic theory and deals with the rule explosion problem, by dividing the process of best service variant selection into two steps. It first independently computes the ideal value of each service QoS parameter, in order to deduce the characteristics of an ideal service variant for a given context state. This step is performed using rules that directly target one QoS parameter, instead of an overall variant satisfaction degree. This reduces the global number of rules and allows processing these rules in several fuzzy controllers, thus reducing rule-based inference processing time. Moreover, this process is applied once per adaptation process, instead of once per adaptation and candidate variant in the classical fuzzy approaches. In a second step, we define a fuzzy similarity method to determine which candidate service variant is the most similar to the previously computed ideal one. This step is applied for every candidate variant but its computation time is linear. The effectiveness of our approach is shown by comparing it with two classical fuzzy control decision making systems.

As future work, several challenges still need to be addressed. For example, we plan to incorporate more generalization in the decision making process.

It can be done by calculating a service composition overall satisfaction degree. This requires the definition of more complex rules to represent the dependencies between the parameters of different variants. Another problem that can be studied concerns the integration of services with the same functional parameters but with some differences in the types of extra-functional parameters. This requires developing a new adaptation strategy to make the decision.

References

[1] N. Paspallis, K. Kakousis, G. Papadopoulos, A multi-dimensional model enabling autonomic reasoning for context-aware pervasive applications, in: Proceedings of 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 08), 2008.

[2] S.-N. Chuang, A. T. Chan, Dynamic qos adaptation for mobile middleware, IEEE Transaction On Software Engineering 34 (6) (2008) 738–752.

[3] K. Kakousis, N. Paspallis, G. A. Papadopoulos, A survey of software adaptation in mobile and ubiquitous computing. enterprise information systems, Enterprise Information Systems 4 (4) (2010) 355–389.

[4] J. Floch, S. Hallsteinsen, E. Stav, F. Eliassen, K. Lund, E. Gjorven, Using architecture models for runtime adaptability, Software, IEEE 23 (2) (2006) 62–70.

[5] G. J. Yaun, Fuzzy sets and fuzzy logic: theory and application, prentice Hall, 1995.

[6] T. J. Ross, Fuzzy Logic With Engineering Applications, Wiley, 2004.

[7] C. Lee, Fuzzy logic in control systems: fuzzy logic controller. i, Systems, Man and Cybernetics, IEEE Transactions on 20 (2) (1990) 404–418.

[8] D. Tsang, B. Bensaou, S. Lam, Fuzzy-based rate control for real-time mpeg video, IEEE Trans. Fuzzy Systems 6 (4) (1998) 504–516.

[9] A. Pitsillides, Y. Sekercioglu, G. Ramamurthy, Effective control of traffic flow in atm networks using fuzzy explicit rate marking, IEEE J. Selected Areas in Comm. 15 (2) (1997) 209–225.

[10] B. Pernici, S. H. Siadat, A fuzzy service adaptation based on qos satisfaction, in: Proceedings of Advanced Information Systems Engineering - 23rd International Conference, CAiSE, 2011, pp. 48–61.

[11] J. Xu, M. Zhao, J. Fortes, R. Carpenter, M. Yousif, Autonomic resource management in virtualized data centers using fuzzy logic-based approaches, Cluster Comput. 11 (3) (2008) 213–227.

[12] D. Gmach, S. Krompass, A. Scholz, M. Wimmer, A. Kemper, Adaptive quality of service management for enterprise services, ACM Transactions on the Web 2 (1) (2008) 8.

[13] L. Zadeh, Fuzzy sets, Information and control 8 (3) (1965) 338–353.

[14] W. Pedrycz, F. Gomide, An introduction to fuzzy sets: analysis and design, the MIT Press, 1998.

[15] T. Kovacs, Genetics-based machine learning, Handbook of Natural Computing: Theory, Experiments, and Applications. Springer Verlag.

[16] R. Yager, D. Filev, Essentials of fuzzy modeling and control, New York.

[17] B. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, et al., Software engineering for self-adaptive systems: A research roadmap, Software Engineering for Self-Adaptive Systems (2009) 1–26.

[18] R. Rouvoy1, F. Eliassen, J. Floch, S. Hallsteinsen, E. Stav, Composing components and services using a planning-based adaptation middleware, in: Proceedings of the n 7th int. symp. on service composition (SC). LNCS, Vol. 4954, 2008, pp. 52–67.

[19] K. Geihs, P. Barone, F. Eliassen, J. Floch, R. Fricke, E. Gjorven, S. H. ans G. Horn, M. U. Khan, A. Mamelli, G. Papadopoulos, N. Paspallis, R. Reichle1, E. Stav, A comprehensive solution for application-level adaptation, Software - Practice and Experience 39 (4) (2009) 385–422.

[20] T. Gu, H. K. Pung, D. Q. Zhan, A service oriented middleware for building context aware services, Journal of Network and Computer Applications 28 (1) (2004) 1–18.

[21] D. Garlan, S. Cheng, A. Huang, B. Schmerl, P. Steenkiste, Rainbow: Architecture-based self-adaptation with reusable infrastructure, Computer 37 (10) (2004) 46–54.

[22] D. Romero, G. Hermosillo, A. Taherkordi, R. Nzekwa1, R. Rouvoy, F. Eliassen, The digihome service-oriented platform, Software - Practice and Experience.

[23] L. Hong, J. Hu, A multi-dimension qos based local service selection model for service composition, Journal of Networks 4 (5) (2009) 351–358.

[24] T. Yu, K.-J. Lin, Service selection algorithms for composing complex services with multiple qos constraints, in: Procedeeings of the Third International Conference on Service-Oriented Computing. LNCS, Vol. 3826, 2005, pp. 130–143.

[25] G. Acampora, M. Gaeta, V. Loia, A. Vasilakos, Interoperable and adaptive fuzzy services for ambient intelligence applications, ACM Transactions on Autonomous and Adaptive Systems (TAAS) 5 (2) (2010) 8.

[26] M. L. Lee, H. Y. Chung, F. M. Yu, Modeling of hierarchical fuzzy systems, Fuzzy Sets and Systems 138 (2) (2003) 343–361.

[27] R. Yager, A new methodology for ordinal multiobjective decisions based on fuzzy sets, Decision Sciences 12 (4) (1981) 589–600.

[28] J. Cao, N. Xing, A. T. S. Chan, Y. Feng, B. Jin, Service adaptation using fuzzy theory in context-aware mobile computing middleware, in: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2005, pp. 496–501.

[29] R. Cheung, G. Yao, J. Cao, A. Chan, A fuzzy service adaptation engine for context-aware mobile computing middleware, International Journal of Pervasive Computing and Communications 4 (2) (2008) 147–165.

[30] M. Huebscher, J. McCann, A survey of autonomic computing degrees, models, and applications, ACM Comput. Surv 40 (3) (2008) 1–28.

[31] E. H. Mamdani, Applications of fuzzy algorithm for control of simple dynamic plant, in: Proceeding of the IEEE, Vol. 121, 1974, pp. 1585–1588.