FUZZY-BASED ARCHITECTURE TO IMPLEMENT SERVICE SELECTION ADAPTATION STRATEGY

Mahboobeh Soresrafil Anari[1], Dr.Afshin Salajegheh[2], and Dr.Yousef Rastegari[3]

[1]Computer Engineering Department, Islamic Azad University, South Tehran Branch,Tehran,Iran
[2]Computer Engineering Department,IslamicAzad University,South Tehran Branch,Tehran,Iran
[3]Computer Engineering Department, Shahid Beheshti University, Tehran, Iran

ABSTRACT

One of the main requirements in service based applications is runtime adaptation to changes that occur in business, user, environment, and computational contexts. Changes in contexts lead to QOS degrade. Continues adaptation mechanism and strategies are required to stay service based applications(SBA) in safe state. In this paper a framework for runtime adaptation in service based application isintroduced. It checks user requirements change continuously and dynamically adopts architecture model. Also it checks providers QOS attributes continuously and if adaptation requirement is triggered, runs service selection adaptation strategy to satisfy user preferences. Thusit is a context aware and automatically adaptable framework for SBA applications. Wehave implemented a fuzzy based system for web service selection unit. Due to ambiguity of context's data and cross-cutting effects of quality of services, using fuzzy would result an optimised decision. Finally we illustrated that using of it has a good performance for web service based applications.

KEYWORDS

Runtime Adaptation, Fuzzy System, Web Service, Service Based Application, SOA framework, QoS attributes.

1. INTRODUCTION

By applying service oriented technologies the WWW is transforming from information based to service based environment thereby business goals are achieved by service coordination and composition. Trend of using SOA technologies is growing because of its features such as loosely-coupled communications, decreasing business/software requirements gap, being standard-based and web-enabled, high reusability and supporting location and technology transparency[1, 2, 3].
In SOA paradigm business requirements transform to business services which are achieved by software services (i.e. web services) in lower layers. Ideally the SBA must be adaptable to match with the user expectations, requirements and preferences[4]. A web service which is represented by a specific service provider may be suitable for one user but at the same time be unsuitable for another user's preferences. Service adaptation strategies such as reconfiguration, reselection, re-composition, renegotiation and sub-process substitution could be used to adapt the state of SBA to new context and requirements.

In online environments, runtime adaptation is a main challenge. Recently several researchers have proposed an approach in which system models, and in particular, software architectural models,

DOI : 10.5121/ijwsc.2013.4401

are maintained at runtime and used as a basis for system reconfiguration and repair[5]. An architectural model of a system is one in which the overall structure of a running system is captured as a composition of coarse-grained interacting components [6]. In [7]an approach for architecture based adaptation is proposed, but this architecture based adaptation isn't suitable for service based applications. The Chisel system [8] introduced a dynamic services adaptation framework which decomposes the particular aspects of a service object into multiple possible behaviours. Whenever the context information changes, the service object will be adapted to use the different behaviours according to the adaptation policy. The Functionality Adaptation method in [9]describes proxy-based context-aware adaptation of service code modules, by which the functionality of a service is adapted based on the estimation of the resource usage required for the execution.

In this paper we present a fuzzy-based adaptation architecture to realise reselection strategy. Proposed architecture consider feature model as input and propose the most proper composite service to run-time layer.

The rest of the paper is organized as follow; in section 2 the overview of architecture and brief description about each layer with the whole functionality process is described. In section 3 tools for implementing this architecture are suggested. The evaluation results of implemented framework based on available technologies and service sets are presented in section 4, and finally section 5 comes with conclusions.

2. OVERVIEW OF APPROACH

We used layered architecture style to decrease the complexity of the problem and the result is a 4-layerd architecture which is illustrated in figure1. These layers includes: 1) *Tasks Layer* 2) *ArchitectureLayer*3) *Services Layer* 4)*Runtime Layer* which we are going to describe each one. In the following the functionality and implementation of this layers will be described.

2.1. Tasks Layer

This layer is an interface between users and SBA. It is responsible for collecting user's requirements, expectations and preferences, and represents these requirements to architecture layer. It analyses user's request and create feature model for it that with user preferences will be sent to architecture layer. Whenever user's requirements changes, this layer analyses the requirements again and updates feature model. So adaptability of this layer according to user's requirements is warranted. Finally this feature model sent to *architecture layer*.

2.2. Architecture Layer

This layer receives feature model from t*asks layer* and is responsible to select architecture model which is the most suitable architecture model for imported feature model. As we know a Software Product Line (SPL) isa families of software systems that share common functionality, but each member also has variable functionality[2].Thus a service oriented product line always has alternative architectures that fit to precise context state[10] and user's requirements. We also conform from this principle and offer for this layer a set of several architecture models, together with an architecture manager. The architecture manager is responsible to determine whether the current architecture model can perform requirements that are represented in feature model[7]. If the current architecture model isn't suitable for feature model, adaptation is repeated again and the most suitable architecture model will be selected. This architecture model is the output of this layer and it sent to s*ervices layer*.

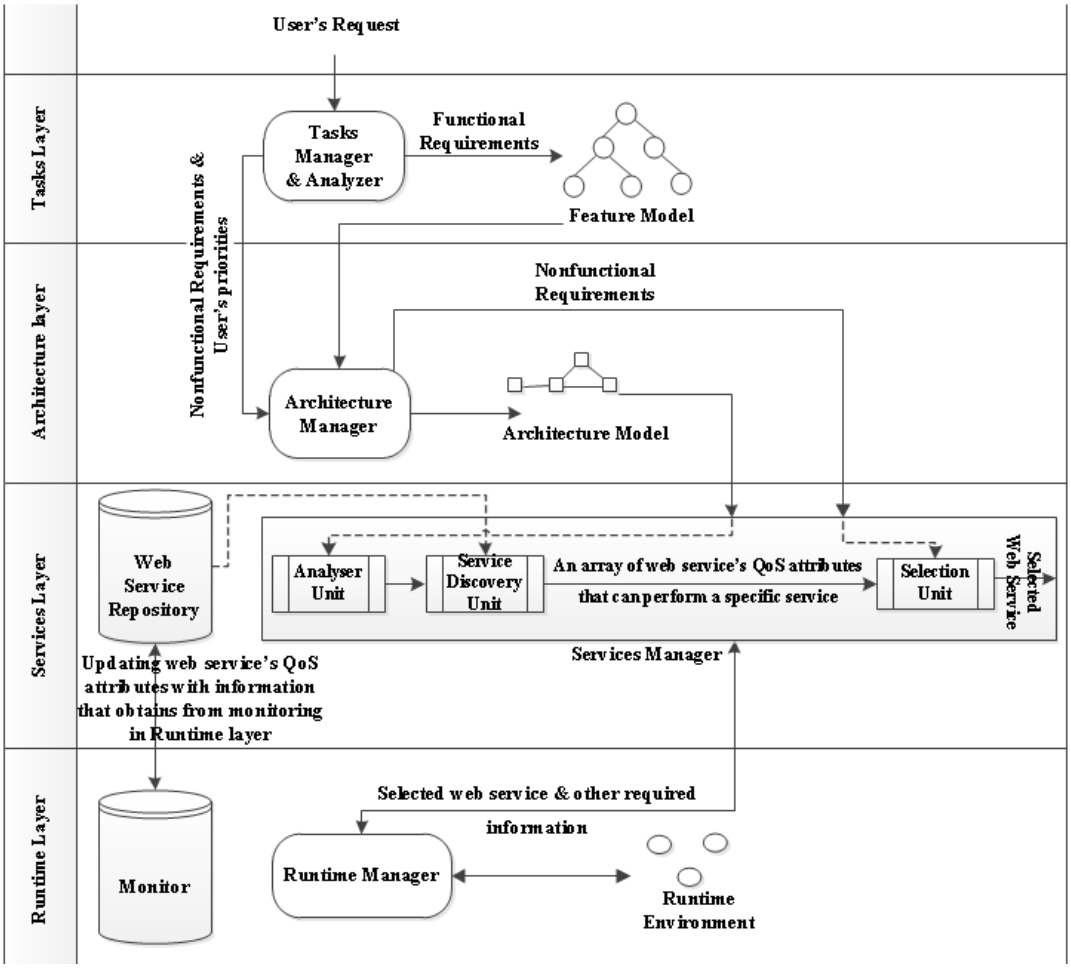

Figure 1. 4-Layers architecture for runtime adaptation in service-based applications

2.3. Services layer

This layer receives architecture model from *architecture layer*. It consists of three units: *Analyser Unit*, *Service Discovery Unit* and *Selection Unit*. The functionality of these units is described below.

2.3.1 Analyser Unit

This unit gets the *architecture model*, analyses it and then creates a composite service model for it which will be passed to s*ervice discovery unit*.

2.3.2. Service Discovery Unit

This unit gets composite service model and has access to repository of web services. This unit for each business service search for candidate services. This unit with information that is received from repository for each service in composite service model identifies a set of web services that can perform the respective service. *Service discovery unit* sends these set of web services with respective QoS attributes to the third part which is named s*election unit*.

2.3.3. Selection Unit

Based on proposed architecture our innovation is presented in Selection Unit. This unit receives two inputs including aset of web services with their QOS attributes from *service discovery unit* and user's preferences which is received from *architecture layer*. It is responsible to selects the most proper web service according to user's preferences.

As we know, context information islargely uncertain and vague, and using fuzzytheory into the adaptation process would make theprocess more flexible and adaptive. So we use a *Fuzzy Evaluation System* for evaluating web services in this unit. Our serviceadaptation model aims at formulating and designing apolicy for web service selection mechanism. This evaluation fuzzy system takes as input a set of web services with theirQoS attributes from *service discovery unit* and user preferences.

2.3.3.1. The Fuzzy Service Adaptation Process

The traditional fuzzy control process consists of three stages: fuzzification, reasoning by inference engine, and defuzzification[11]. In the following paragraph we describe these stages with more details.

2.3.3.2. Fuzzification

Infuzzification stage, predefined membership functionsfor each linguistic variable are used to determine the degree of membership of a crisp value for the linguisticvariable[11].We represent a linguistic variable for each of context in formation by in this step. Each linguistic variable is represented by a predefined application-related membership function.

Sure we can choose different QOS attributes for this fuzzy system. But we choose the most important QOS attributes namely Availability, Reliability and Performance for our discussion.We define 4 ranges for these attributes. These levels consist of: Poor, Average, Good and Very Good which have numerical ranges. The definition of these ranges, that gets by consultations with experts, shown in figure 2. User can selects one of the three levels for availability, reliability and performance. These 3 options are: Average, Good and Very Good. This set is the second input parametersof fuzzyevaluation system.

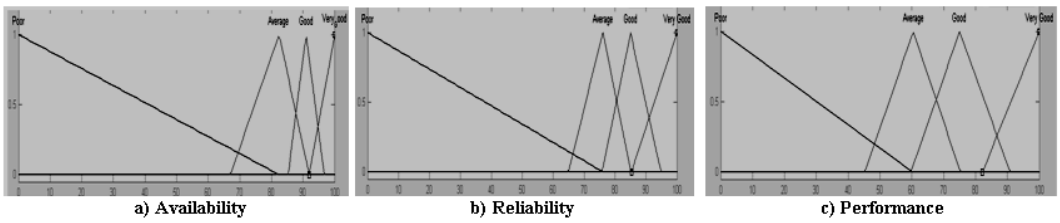

Figure 2. Definition of linguistic variables: (a) Availability, (b) Reliability and (c) Performance in *Evaluation Fuzzy System*. Ranges defined by consultation with experts.

2.3.3.3. Reasoning by inference engine

By using the membership functions we translate the input context crisp value into a set of pairs, each consisting of a linguistic value and a membership degree for that value[11]. The inference engine refers to the predefined fuzzy rule base to derive the linguisticvalues for the intermediate and output linguisticvariable. So we must discuss about Fuzzy rules for this system.

We must add any composition of input variables to rules engine. As we said previously we have three variables: Availability, Reliability and Performance. So this engine rules has seven rules (the composition of 3 variables is: $\binom{3}{3}+\binom{3}{2}+\binom{3}{1}=7$). The weights of these rules must be different based on user preferences. The sum of rule's weight is 1. The rule that is the most adaptable with user's preferences (the rule that in it, all variables have the same value as the user's preferences)has the highest weight and other rules have lower weight. Whereas, these rules must select the best web service per user's preferences, and user's preferences aren't static, these rules aren't static too. When user changes the value of the parameters(Availability, Reliability and Performance)rule's parameters values change consequently. So these rules aren't static and as soon as user selects the value of his preferences, fuzzy rules will be updated automatically.

In Figure 3 fuzzy rules for case that user selects following quality levels are shown:
$\{(Availability = Good),(Reliability = Good),(Performance = Average)\}$

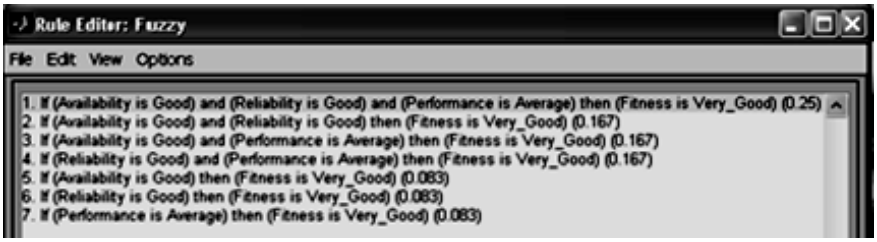

Figure 3. Fuzzy rules in case that user selects these options:*{(Availability=Good), (Reliability=Good),(Performance=Average)}*.

Also in Figure 4 fuzzy rules for case that user selects following options are shown:
$\{(Availability = Average),(Reliability = Average),(Performance = Average)\}$

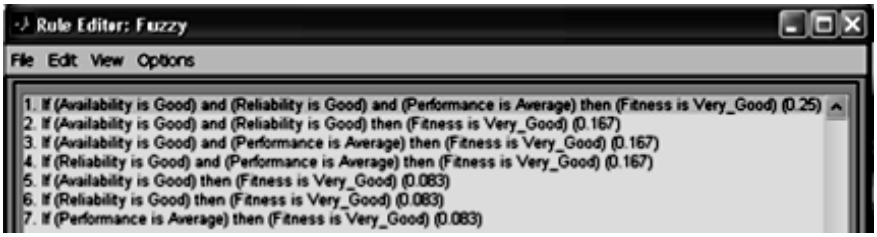

Figure 4.Fuzzy rules in case that user selects these options:*{(Availability=Good), (Reliability=Good),(Performance=Average)}*

2.3.3.4. Defuzzification

Finally this *fuzzy evaluation system* only has one output that we named it Fitness. During the inference process in *fuzzy evaluation system* each rule will be assigned a fitness degree that indicating to what degree the web service is suitable for being used under the current context situation and user preferences. *Fuzzy evaluation system* sums these fitness degrees together and assigns it as fitness of this web service in output. So fitness specifies the adaptability of each web service to user's preferences. The output of *fuzzy evaluation system* is an array of numeric data, that each of them specifies the adaptability of respective web service with user's preferences. After the fuzzy evaluation is done, the largest value of this array is respective to the most adaptable web service with user preferences.

In Figure 5decision surface of fuzzy evaluation system for two cases is shown.

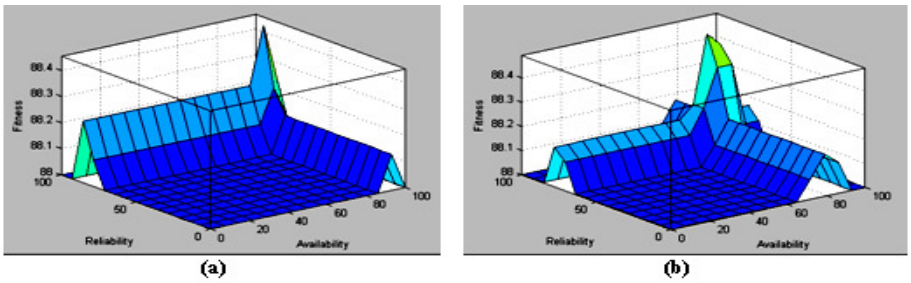

Figure 5. The decision surface of fuzzy evaluation system for two cases: (a)When user selects *{(Availability=Good), (Reliability=Good)}*.(b)When user selects *{(Availability=Average), (Reliability=Average)}*

As described in previous section, when user changes his preferences, rule's parameters change automatically and different web service selected the best web service. Figure 6 shows this subject. In this figure in (a) user selects{$(Availability = Average)$,$(Reliability = Average)$, $(Performance = Average)$}.In this case among 2500 web services a web service with index 656 is the winner of this competition. Also in (b) user selects {$(Availability = Good)$, $(Reliability = Average)$, $(Performance = Average)$} and In this case among the same web services, a web service with index 360 is the winner of this competition.

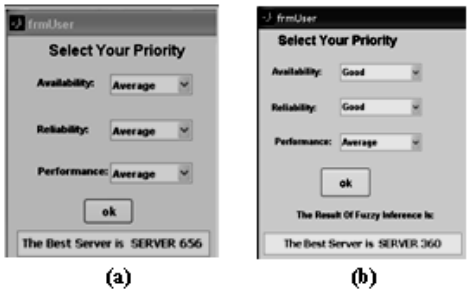

Figure 6. When user changes his preferences fuzzy evaluation system selects different web service as best web service.

With this implementation, rules updated dynamically and in result per user's preferences and web services QoS attributes each time different web service selected as a best web service. The respective service provider is selected as the output of the selection unit and is responsible for running service.

Finally, *services layer* makes necessary decisions and specifies that each of services in composite service model must be performed with which service provider. Then it sends instructions to *runtime layer* for executing the service.

2.4. Runtime Layer

This layer is responsible for monitoring runtime environment contexts and executing the services with service providers that previous layer identified. It consists of system itself, together with its operating environment (such as networks, communication links, service providers, etc). We

considered feedback loop to update service repository based on provider's run-time service quality level.

3. IMPLEMENTATION

We implement these layers separately. Infrastructure currently supports suitable tools for implementing Tasks Layer, Architecture Layer and Runtime Layer, so we only suggests suitable tools for implementing these layers and focus on implementing Services Layer. Also, in Services Layer, suitable tools exist for implementing, analysing and discovering units. So we only suggest tools for implementing these parts and focus on implementing Selection Unit.

In the next section we first describe implementing of Tasks, Architecture and Runtime Layers andthen focus on Services Layer and describe methods for implementing it.

3.1. Tasks Layer

As we discussed previously, this layer receives user's request and create feature model for it. For this layer we plan to take advantage of existing strategies for representing tasks. We can use tools such as Business Modeller[12]from IBM Corporation for implementing it.

3.2. Architecture Layer

As we know, this layer receives features model from Tasks Layer, and create architecture model from it. We can use products such as AcmeStudio [7, 13, 14]for implementing this layer.It makes architectural model at runtime, namely if user's requirements changes during runtime, it updates architecture model without breakdown in representing service to user.

3.3. Runtime Layer

This layer is responsible for surveying runtime environment and sends information about it to Services Layer. In other hand, it must monitors system's runtime status. Things which must be monitored consists of service providers and theirs runtime environment such as networks, processors, I/O devices, network paths, etc.

We can use the existing tools which are used for monitoring. One of them is REMOS. It monitors runtime environment such as networks, servers, etc[7, 15, 16]. After collecting data, these data are analysed, the result is information about service provider's QOS attributes that sent to Services Layer with messages, the Repository updates itself with these information and hereby Services Layer, know the status of each service provider.

3.4. Services Layer

3.4.1. Analysing Unit

This part receives architecture model and is responsible to create composite service model from it. We can use methodologies such as SOMF for implementing this part. It can employ during various stages of the software development life cycle[17, 18]. It provides a technology-independent notation that encourages aholisticview of enterprise software entities, treated as service-oriented assets, namely services[17, 18]. Also tools such as Web Sphere Business Modeler from Oracle Corporation[19] can employed for implementing this unit.

3.4.2. Discovery Unit

This part is responsible for identify web services that present a specific service. Many tools and methods are introduced for this job. We can use one of these methods, such as Rondo[20, 21] Clio[22, 23], Coma[22, 24], etc for implementing it.

3.4.3. Selection Unit

This part receives two inputs, a set of service providers with respective QOS attributes and a set of user's preferences. This partselects the most adaptable web service among a set of web services according to user's preferences, and set it as output. We described the implementation of this part previously.

4. EVALUATION

In order to evaluate the complexity of this approach, we have implemented this evaluation fuzzy system in Matlab application on a windows machine with Pentium 4, 3GHz CPU and 3GB memory. It was assume that each service has n candidate web service. Q shows the number of quality attributes which we consider 3 quality attributes: Availability, Reliability and performance and s is the number of atomic service in a composite service.

We obtain value of quality service attributes from dataset [25]. In this dataset amount of QOS parameters of some real services is measured. In order to use this dataset, we consider them as candidate services for performing a task.

Figure7shows time of fuzzy evaluation when n, the number of candidate service increases from 1 to 2500. In this graph q=3 and s=10. This graph approximately is linear but has few jumps. Thus this Fuzzy system has O(n) complexity, with gradient 0.0002.

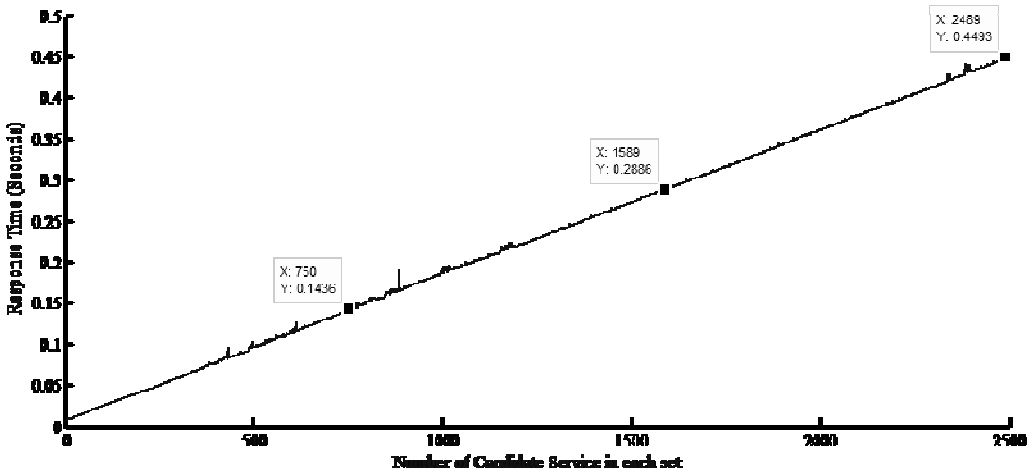

Figure7.Complexity evaluation time per number of candidate services

Figure8 shows time of fuzzy evaluation when the number of atomic services increases in a composite service. In this graph q=3 and n=2500. S, the number of atomic service in a composite service increases from 1 to 50. This graph is a line with gradient about 0.045.

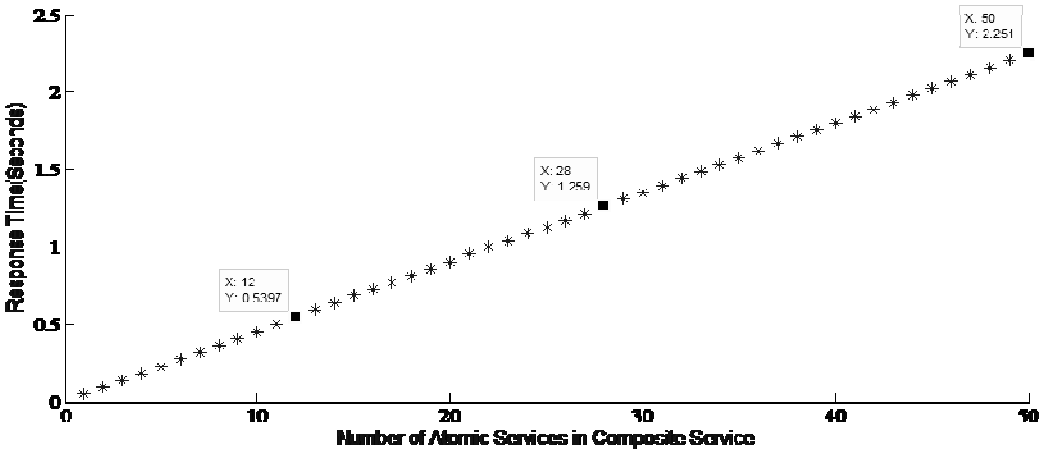

Figure 8. Complexity evaluation time per number of atomic service in a composite service

Figure9 shows the time of fuzzy evaluation when the number of QOS attributes changes. In this graphs, s= 10 and we change n, the number of candidate service, from 1 to 2500. Then calculate the evaluation time when the number of QOS attributes is 1, 2, 3 and 4. Then measure the time of evaluation and draw the evaluation time in graphs. As shown in figure9 the time of evaluation approximately reduplicated for each increase in number of QoS attributes.

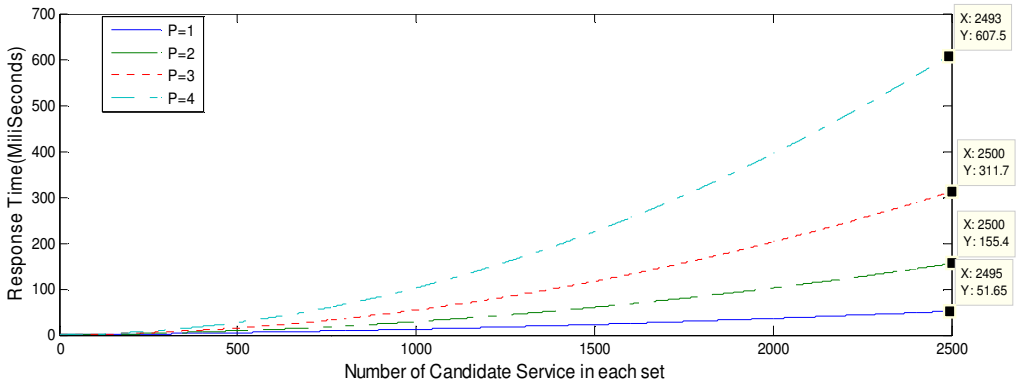

Figure 9. Complexity evaluation time per number of QOS attributes

3. CONCLUSION

In this study we have suggested a 4-layers approach for runtime adaptation for SBA systems. We use fuzzy system to select best service provider according to user's preferences in Services Layer of this approach. It is a fully automatic and dynamic approach for service selection adaptation according to user's preferences and context's data. We have studied this approach for evaluating its complexity.The results show that this fuzzy system has enough capability to be implemented. Since complexity of this system is O(n) with a small gradient, it has enough efficiency for selecting best service provider according to user's preferences.

REFERENCES

[1] L'ecu'e, Freddy & Silva, Eduardo & ferreira Pires, Lu'is, "A Framework for Dynamic Web Services Composition", France : Centre for Telematics and Information Technology University of Twente , The Netherlands. France Telecom R&D. 1.

[2] H.ter Beek, & Maurice, Gnesi & Stefania & N.Njima, (2011) ,"Product Lines for Service Oriented Applications - PL for SOA".

[3] Kumar, Nitin,(2013),"Benefits on Service Oriented Architecture – SOA", is available at: http://www.javacodegeeks.com/2013/03/enterprise-benefits-on-service-oriented-architecture-soa.html.

[4] Schou, Saowanee, "Conceptual Service Architecture For Adaptive Mobile Location Services On The Next Generation Wireless Network". Ubiquitous Computing and Communication Journal.

[5] Oreizy, P & et al, (1999),"An Architecture-Based Approach to Self-Adaptive Software"., pp. 54-62.

[6] Shaw, M & Garlan, D. ,(1996), "Software Architectures: Perspectives on an Emerging Discipline".

[7] Cheng, Shang-Wen & et al ,"Software Architecture-Based Adaptation for Pervasive Systems", Verlag Berlin : pringer. INRIA Lille-Nord Europe.

[8] Keeney, J & Cahill, V,(2003), "Chisel: a policy-driven, contextaware, dynamic adaptation framework", Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks, pp. 3-14.

[9] Wai-Man Kwan, Vivien, Chi-Moon Lau, Francis & Li Wang, Cho,(Oct.2003), "Functionality adaptation: a context-aware service code adaptation for pervasive computing environments". Proceedings of IEEE/WIC International Conference on Web Intelligence. pp. 358 – 364.

[10] Parra, Carlos, Blanc, Xavier & Duchien, Laurence, "Context Awareness for Dynamic Service-Oriented Product Lines", France : s.n. INRIA Lille-Nord Europe.

[11] Cao, Jiannong, et al, (2005)"Service Adaptation Using Fuzzy Theory in Context-aware Mobile". 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.

[12] It's available at: http://www.businessgenetics.com/Products/BusinessProcessModeler.aspx.

[13] Garlan, David, Schmerl, Bradley & Chang, Jichuan. s.l,(2001), "Using Gauges for Architecture-Based Monitoring and Adaptation", : Computer Science Department, p. 690.

[14] It's available at: http://www.cs.cmu.edu/~acme/AcmeStudio.

[15] Lowekamp, Bruce & et al "A Resource Query Interface for NetworkAware",. s.l. : School of Computer Science.

[16] cheng, Shang Wen & et al, (2002). "Software Architecture-based Adaptation for Grid Computing", The 11th IEEE Conference on High Performance Distributed Computing.

[17] Service Oriented Modeling framework TM in Enterprise Architect. It's available at: www.sparxsystems.com/somf.

[18] Service-Oriented Modeling Framework (SOMF) version 2.1. It's available at: www.sparxsystems.com/downloads/whitepapers/SOMF-2.1-Discovery-and-Analysis-Model-language-Specifications.pdf.

[19] It's available at:http://docs.oracle.com/cd/E21764_01/doc.1111/e17368/chapter.htm.

[20] Letz, Carolin,(2007),"Web Service Detection in Service-oriented Software Development: A Semantic Syntactic Approach", p. 85.

[21] Melnik, Sergey, Rahm, Erhad & Bernstein, Philip A.(2003)," Rondo: A Programming Platform for Generic Model", pp. 193-204.

[22] Letz, Carolin, (2007). "Web Service Detection in Service-oriented Software Development: A Semantic Syntactic Approach", p. 86.

[23] Fagin, Ronald, et al. "Clio: Schema Mapping Creation and Data Exchange".

[24] Do, Hong-Hai and Rahm, Erhad. "COMA - A system for flexible combination of schema matching approaches". Proceedings of the 28th VLDB Conference. 2002.

[25] Al-Masri, eyhab & H.Qusay, Mahmoud. It's available at: http://www.uoguelph.ca/~qmahmoud/qws/dataset/.