Vol. 5(14) Jan. 2015, PP. 2015-2029

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Efficient Web Services Composition with Fuzzy–Genetic Approach

Hossein Momeni[1] and Fatemeh Mahdavi[2]

[1]Computer Engineering Department, Golestan University, Gorgan, Iran

[2]Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

***Corresponding Author's E-mail:** h.momeni@gu.ac.ir

Abstract

Nowadays, Service-oriented architecture is known as a hopeful and effective solution for making the applications organizations which of business operations support as a set of well-defined service. Web services selection and composition are key activities which are done in of the various phases in the life cycle of service-oriented architecture. Selection a valuable web services due to existing many services and also having variety of criteria in the production process automatically and is difficult the selection work. Automatically production and selection web service is one of the important key activities in the implementation a successful service-oriented projects. Identify existing approaches are frequently prescriptive and experience-based architect. Thus it is possible lead to be non-suitable designs, reduce resulting performance, reducing of scalability and complex dependence between services. In this paper, we present an automatically method for identify business services. Using genetic algorithm and fuzzy logic can be production automated web service based on qualitative and quantitative characteristics. In this paper, architect can use metrics to measure characteristics of web services then is assured the abstraction level of web services. Genetic algorithm is for the automatic identification of web services and improved role of architect. Fuzzy logic selects web services based on user preferences. The user view has direct effect in rank for composition of service. Therefore user selected good composition of web service with our ideas. This approach in web services selection and composition compare to previous approaches have upper efficiency and improvement.

Keywords: *Service-oriented architecture, Identification of service, Selection of service, Composition of service, Genetic algorithm, Fuzzy logic.*

1. Introduction

A web service is software which uses XML to transfer data between other software via internet protocols. A web service has properties which make it distinct among other technologies. These properties are: First, a web service is programmable. Second, it is established based on XML. Third, a web service is self-descriptor and identifiable [8]. Many descriptions and web service framework for addressing web service transactions and synchronization are offered to be standardized. Each solution has targeted a particular problem which doesn't meet other requirements. Therefore, it is essential to identify key elements of SOA, restrictions in supporting service composition, verification, and composition automating [18]. The structure of web service basic is enough for simple interactions. If one web service need another web service and several web service must combined together. This operation called complete services and the processing of complete services also called composition service. They need to several services all the time composition. Services are basic requirement for developing the gold of organization to answer what the customer requirement. Therefore, the web services are good technology for dynamic of effective commercial cooperation, at

Article History:
Received Date: Dec. 07, 2014
Accepted Date: Feb. 20, 2015
Available Online: Feb. 25, 2015

result it can cause connected of the organization and use of the data the necessary use of web services to cause until single web services combined together and make a complex service. The ways for combined web services often are basis of workflow and artificial intelligence, and the main ways for combined services include:

a. Static composition

b. Dynamic composition

There are fixed rulers in the static composition that depended to work and activity in future. The problem of this web service is that cannot give framework for making the complex process. In dynamic composition, we can reduce the time of web services composition. Our gold in this paper should cover weakness of before work. Therefore we must use of genetic-fuzzy algorithm that include:

a. Use of short memory

b. Find intellectual solutions in big spaces and infinite

c. And reported algorithm and it can produced the best solutions.

At the end, increase the numerous of web services that have same operation but they do not have some qualities. Quality Service involved nonfunctional features like execution cost, execution time, availability, rate of successfully in execution and security. Many services have much complex; therefore they must do by composition of web services. Also choice and the composition of web services are the kinds of important studies, in this paper. To offer the algorithm of the best combined web services according to genetic-fuzzy algorithm.

In this algorithm, produced chromosome, selected chromosome, cross-over and mutation have an optimal result. The rest of this paper is organized as follows. In Section 2, the basic concepts of business process and multi-objective optimization are defined. In Section 3, the most related work is briefly reviewed. In Section 4, the service identification and composition problem is introduced and formally defined and a method in order to solve the service identification and composition problem is proposed. Then, it has been detailed in Section 5 by solving the problem with genetic algorithm (GA) and logic fuzzy and a case study has been conducted. In Section 6, evaluate the proposed solution. Finally, the conclusions are discussed in Sections 7.

2. Background and definition

In this section, we briefly review the background and definitions in service identification and composition.

2.1. Web service composition

Web services are applications on the web, independent of their context, which can be called and executed. The structure of a web service is sufficient to implement simple interaction between a client and a web service, but as applications are becoming more and more complex, there is also a need for more complex web services [17]. Web service composition is a process which several web services are incorporated to create value-added services [12]. This process begins with a request from the client to use a composite service. After this process is reviewed, the service will be selected from available services. Then a description of the composite service is provided by an appropriate language. And finally the execution engine runs according to the definition of composite service [19, 15].

2.2. Genetic algorithm

Genetic algorithm is a kind of random search. A genetic algorithm such as local search starts with sets of random mode which is called the population. Each mode or individual is displayed as a string of a finite number of alphabets. These modes in genetic algorithm are called chromosomes which consist of several genes. Genetic algorithm is an evolutionary algorithm to search in the vast space [14].

2.3. Fuzzy logic

Fuzzy logic is a multi-valued logic that allows values to be defined in logic values such as true/false, up/down, and yes/no. In this theory, the membership is indicated by the function u(x) which x represents a specific member and u is the degree of x membership in the set and is determined by a value between 0 and 1. This is shown as relation (1) as follows:

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{1}$$

Fuzzy logic can be applied through rules which are called fuzzy operators. These rules are usually based on relation (2) that is shown as follows [5].

IF variable IS set THEN action (2)

2.4. Definition

In this section, the relevant terminologies and its implications are defined.

Definition 1 (Business process): A business process (BP) is a set of logically related activities which are designed to achieve pre-defined business goals [16].

Definition 2 (Business Activity): A Business Activity is an atomic task in a business process [16].

Definition 3 (Data Flow): Defines the interaction between activities, for example the outputs of an activity may be used as inputs of other activity [11]. Each data flow contains one or some business entities which are exchanged as a message between activities.

Definition 4 (Business Entity): A Business Entity (BE) is a dominant data unit. Each BE has a relative complexity so that the designer can estimate it base on his/her experience [11].

Definition 5 (Multi-Objective optimization): A Multi- Objective optimization problem is defined as follows [3]:

$$\text{Min } f(x) = \{ f_1(x), f_2(x), \ldots, f_M(x) \}$$

$$X \text{ € } x^{nx} \tag{3}$$

$$\text{S.t.} g(x) \geq 0 \text{ , } h \geq (x) = 0$$

Where x is the vector of decision variables bounded by the decision space, X and f are the set of objectives to be minimized.

3. Related works

In this section, we briefly review the most related work in service identification and composition. In [20, 4] Zang introduced a distributed architecture called MOVE. The architecture uses a great scale to optimize the extraction of sub-ontology. The architecture was developed to address heath issues. By analyzing the semantic similarities between interfaces of individual services and the quality of service (corresponding QoS), possible combinations are obtained. Thus, these compounds are ranked

and presented according to two dimensions: Mrissa et al offered a dimension for web service composition based on semantic contents. This combination is done based on the understanding of concepts of interaction and the basic capabilities of the service. Service composition techniques are controlled by end users. So the set of testing system capabilities are completely limited. As a result, the system doesn't have the monitoring tool to run the composite service. This suggests that the failure is not assessed during the runtime.

The approach followed by enTish [2] is somewhat different from typical composition platforms. Services are typically created on the fly to realize client requests. Anyway, most frameworks are based on the assumption that first the business process has to be created. For enTish, a different architecture is needed, since client requests are expressed in a declarative way using formal languages. The declarative approach consists of two phases: the first phase takes an initial situation and the desired goal as starting point, and constructs generic plans to reach the goal. The latter one chooses one generic plan, discovers appropriate services, and builds a workflow out of them. The first phase is realized using PDDL (Planning Domain Definition Language) and estimated-regression planning as used in XSRL (XML Web-services Request Language), which must provide machine-readable semantics and specify the abstract service behavior. The second phase may be realized by using existing process modeling languages, such as BPEL.

In [18 ,9] authors present Formal methods such as automata theory, algebra process, finite state machines, and Petri Nets can be used to model and implement a web service composition. The methods support the formal modeling and evaluation service composition, and can improve the accuracy in certain situations, and make it easy to automate the composition. Thus, this technique does not provide a method for web service. On the other hand, this approach imposes a series of requirements which are not true but must be implemented. They present a framework that automatically constructs a Web service composition schema from a high-level objective. The input objective is fed to an abstract workflow generator that attempts to create an abstract workflow (written in BPEL) that satisfies the objective, either by using already generated workflows or subsets of them that are stored in a repository or by performing backtracking to find a chain of services that satisfy the objective. The abstract workflow is then concretized, either by finding existing services through a matchmaking algorithm 38 that matches inputs and outputs and binding them to the workflow, or by recursively calling the abstract workflow generator if no service can be found for an activity.

In [7] Azevedo et al have presented a systematic method for service identification in three phases which are: "Selection of Activities", "Identification and Classification of Candidate Services" and "Consolidation of Candidate Services". In the first phase, a set of activity is taken as input and then those activities which can be done automatically are selected. In the second phase a set of heuristic has been defined which are applied on the activities selected from first phase. The outputs of this phase are candidate services which are obtained by syntactical and semantic analysis of the process model and making use of mentioned set of predefined heuristics. In the third phase, the obtained services are consolidated. Although in this paper a systematic approach has been proposed for service identification, most of proposed heuristics cannot be measured automatically. For example one of the heuristics is that the candidate services must be identified from business rules. But this is a general guideline and there is not any explanation about how to do it and especially the way of its automation while this is an enormous and comparatively complex task. Also this paper has not used any technical metric to ensure appropriateness of service abstraction level.

4. Proposed approach

The purpose of identifying services is to create a set of candidate services and operations associated to them . Service modeling is one of the most common discussions of information technology executives and architects, and it is going to be one of the SOA challenges. Typical challenges are as follows [10]:

Challenge 1: How do we establish criteria for measuring quality attributes like granularity and reusability?

Challenge 2: How do we identify appropriate candidate services?

Challenge 3: How do we prioritize services?

In this paper we will answer to the three aforementioned major challenges in service-oriented modeling phase. We will define a set of step-by-step automated process to give an answer to the raised challenges. In the following we will explain the proposed service identification and composition method.

The Figure 1 shows the method in a format of a process model. This process model includes three phases which are described later in this section and deal with the mentioned challenges in service-oriented modeling and contains a set of tasks which clarify them respectively.

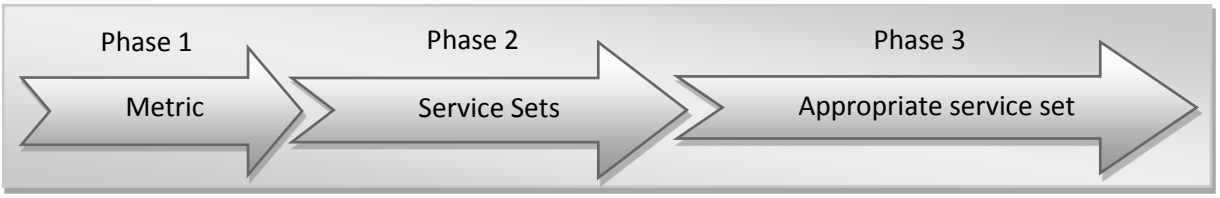


Figure 1: The proposed method

4.1 Metric definition phase

In this phase a set of business goals are taken and set of appropriate metrics are defined or selected. Thus achieving a set of appropriate metrics from business goals is the major purpose of this phase. First, a set of important quality attributes which cover business goals are selected. Since different quality attributes satisfy different business goals, it is required that the designer selects a set of appropriate quality attributes aligned with those goals. For example, some of enterprise goals can be listed as follows [10]:

- Be agile to adapt quickly to new opportunities and potential competitive threats
- Be first to market with innovative services for its customers
- Provide streamlined business processes to reduce operating costs

Specific business goals propagate their required quality attributes, for example agility and first to market business goals mean that Adaptability, Interoperability, Scalability, and Extensibility quality attributes are the major drivers for application architecture and service development, and it is possible that the importance of these attributes differ according to business goals [13]. In the next step, appropriate metrics are defined for each quality attribute so that they can be measured automatically. The key output of this phase is a set of metrics which are used in the second phase. The first phase of the proposed method deals with the first challenge in service-oriented modeling and contains a set of tasks which answer this challenge clearly.

4.2 Service set identification phase

The ultimate goal of this phase is automatic identification of candidate service set. Business process models together with business entities and also the defined metrics in the first phase constitute the inputs of this phase. In this phase, characteristics required for the modeling of the problem according to definition 5 are derived from business process model; these characteristics

themselves constitute a specific process model. Then the problem of identifying services is modeled as a genetic based multi-objective optimization algorithm. By setting the parameters of GA, implementation is done and then executed on a specific model. In an evolutionary process, this algorithm will produce a set of non-dominated solutions as a Pareto set. The key output of this phase is a Pareto set of solutions together with value of metrics related to each solution. The second phase of the proposed method deals with the second challenge in service-oriented modeling and contains a set of tasks which answer to this challenge clearly.

4.3 Appropriate sservice set selection phase

In this phase, first, according to importance of quality attributes, a set of fuzzy rules together with their membership functions are defined.

- Quality of service Composite

The quality of service, according to qualitative criteria (cost, response time, availability, and reliability), for initial services of service quality vector is defined as relation (4, 5). Some of these functions are shown in Table (1).

$$Q(s) = (qcost(s), qRt(s), qAv(s), qRe(s)) \quad (4)$$

$$Q(s) = \sum(qcost(s), qRt(s), qAv(s), qRe(s)) \quad (5)$$

TABLE 1: MASS FUNCTIONS

QoS Attr	
Cost	$\sum_{1=1}^{n}\sum_{j=1}^{m} c_{ij}$
Time	$\sum_{1=1}^{n}\sum_{j=1}^{m} t_{ij}$
Availability	$a_{ij} = \frac{req_{ij}}{tot_{ij}}, tot_{ij} \neq 0,$ $\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}}{n} > 0$
Reputation	$q_{res}(S) = \frac{\sum_{i=1}^{n} k(s_i)}{n}$

By using the above mass function, the vector of a composite service with an execution map of P is defined as relations 6, 7 and 8:

$$Q\ (p) = (Qcost\ (p), QRt\ (p), QAv\ (p), QRe\ (p)) \quad (6)$$

$$Q\ (Total) = \sum_{i=1}^{n}(qCost(si) * qRt(si) * qAv(si) * qRe(si)) \quad (7)$$

(8)

$$Q(T)=\sum_{i=1}^{n}(\sum_{i=1}^{n}\sum_{j=1}^{m}C_{i,j},\sum_{i=1}^{n}\sum_{j=1}^{m}T_{i,j},\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}A_{i,j}}{n}>0,\frac{\sum_{i=1}^{n}K(S_i)}{n})$$

After identifying the set of services, the services must be ranked using fuzzy logic and according to user requirements. After identifying the needs and preferences of their users, IF-THEN rules are implemented. This approach can be a base for ranking the quality vectors of execution plans. More important rules are ranked higher. The third phase of the proposed method deals with the third challenge in service-oriented modeling and contains a set of tasks which answer to this challenge clearly.

5. Modeling the service identification and composition with GA-Fuzzy logic

Now we describe the scenario of a web service in a travel agency. Today, various models and languages have been proposed to describe a service; the state diagram is one of the most common ones. Diagram of modes are shown in Figure (2):

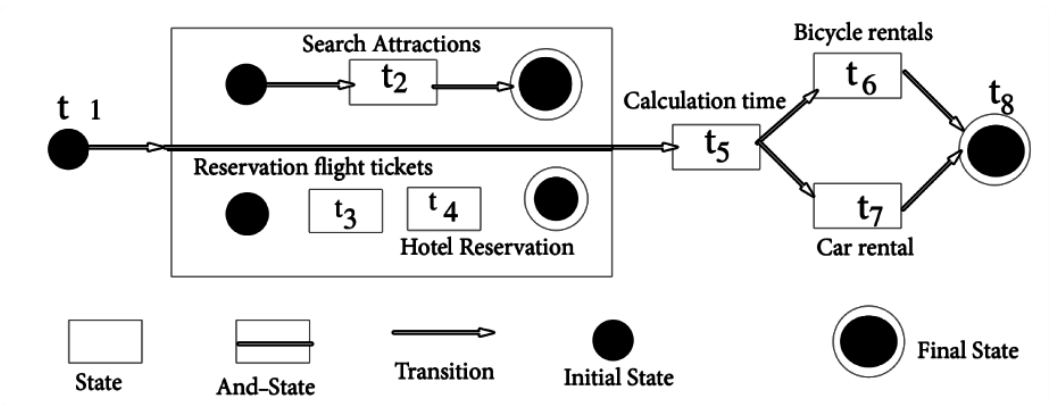


Figure 2: State chart of a composite service Travel Planner

5.1 Modeling of phase 1

We consider three business goals as inputs in this case study: Agility, Ease of Future Enhancement and High Value to Business. A set of design metrics which satisfy business goals should be selected or designed in the first phase. According to the fact that many work has been done in the field of defining design metrics [1] we have chosen four design metrics. These metrics are: cohesion, coupling, business entity convergence, and granularity. Considering the mentioned business goals, quality attributes which cover these goals and also the relationship between these metrics and quality attributes are shown in Table (2).

TABLE 2: RELATIONSHIP BETWEEN BUSINESS GOAL, QUALITY ATTRIBUTE AND METRIC

Business Goal	Quality Attribute	Metric
Agility	Coupling	V_{COPL}
Agility	Granularity	V_{GRANU}
Ease of Future Enhancement	Cohesion	V_{CHOS}
High Value to Business	Entity Convergence	V_{CONVE}

5.2 Modeling of phase 2

The metrics VCOPL, VGRANU, VCHOS and VCONVE, which had been defined in the first phase, constitute the first input of this phase. Two other inputs of this phase are business processes and

business entity weights. Figure 3 illustrates process model of this scenario. There should be a process model which is mapped to the business environment according to the mentioned scenario. Parameters of GA were determined as follows for fast convergence: Population size = 500, Generation span = 100, Crossover rate = 0.95, Probability of mutation = 0.05.

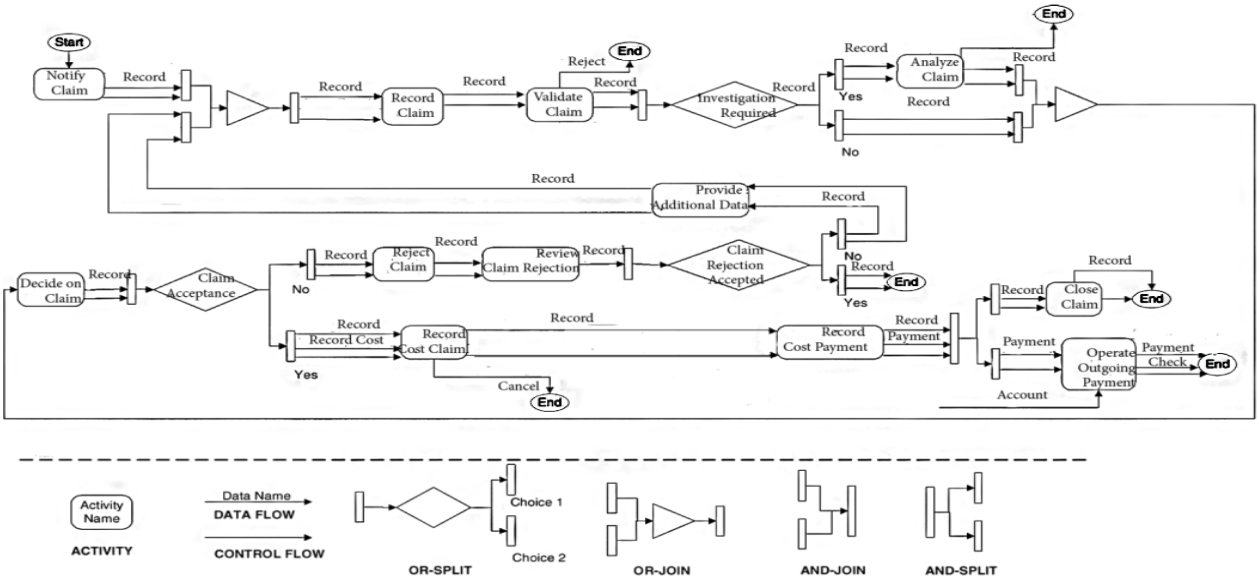


Figure 3: Administering Travel Planner

- **Representation and initial population creation**

In order to apply the GA to a particular problem, we need to select an internal string representation for the solution space. The choice of this component is one of the critical aspects to the success or failure of the GA regarding the problem of interest. In our method, we represent each chromosome as a sequence of service activity. Generally a business process can be partitioned at most to n service, where n is the number of existing activities in business process. In other words, activities in a business process can be placed in one service, two services, three services, and or n separate services. Each chromosome which is representative for a service set contains n genes. Activities inside genes are placed in at most n services. Initial population is obtained by creating specific number of these chromosomes.

- **Crossover operator**

One of the unique and important aspects of the techniques involving GA is the important role that recombination (traditionally, in the form of crossover operator) plays. In this paper, we propose a new crossover operator, called Service-Crossover. An offspring C1 is created from two parents, P1 and P2 as follows:

1. Given a business process with n activities, randomly select a number i, (where i = 1, 2… n-1).

2. Randomly select i activities which are the activities to be passed on from parent to offspring per parent.

3. One gene is copied directly from P1 to C1 if it contains one of previously selected i activities.

4. Remained genes are copied from P2 to C1 if it does not contain any of previously selected i activities.

- **Mutation**

After cross-over, chromosomes are mutated. Mutation prevents the algorithm from being trapped in a local optimal. Mutation plays the role of retrieving lost genetic information by randomly

distributing genetic information. In this algorithm, to perform mutation operator, two services, Si and Sj, are selected randomly. Then two activities, Ai and Aj, which are in Si and Sj services respectively, are selected to exchange. GA is executed as many as the number of generation span and optimal service set is obtained through this evolutionary process. One of the chromosomes of the above business process in population is as follows:

(S1, Notify Claim)(S2, Record Claim)(S1, Validate Claim)(S3, Analyze Claim)(S1, Decide on Claim)(S2, Reject claim)(S3, Provide Addition Data)(S4, Review Claim rejection)(S3, Offer Benefit)(S4, Prepare Claim Discharge)(S3, Record Benefit Payment)(S2, Close Claim)(S1, Operate Outgoing Payment).

As above, 500 chromosomes are created and algorithm enters an evolutionary process and finally produces a solution as Pareto solution set.

- **Objectives of the fitness function**

In our issue, four major objectives among quality features are considered: Minimizing the cost, minimizing response time, maximizing the availability, maximizing reliability. Five Pareto solutions have been shown in Figure (4). Genes inside chromosomes are in the format of (Service, Activity). Moreover, the average value of each metric has been shown for the Pareto sets.

Non-Domination Solution	**Granularity**	**Coupling**	**Cohesion**	**Convergence**
A_1=Notify claim, A_2= Review claim Rejection, A_3=Record cost claim, A_5=Record Claim, A_6=Record cost Payment, A_7=Validate claim, A_8=Analyze claim, A_9=Decide on claim, A_{10}=Reject claim, A_{11}=Close claim, A_{12}=Prepare Payment, A_{13}=Provide additional date				
$(S_1,A_1)(S_2,A_2)(S_2,A_3)(S_1,A_5)(S_2,A_6)(S_1,A_7)$ $(S_1,A_8)(S_1,A_9)(S_1,A_{10})(S_2,A_{11})(S_2,A_{12})(S_1,A_{13})$	6.5	3	1.7	4.2
$(S_1,A_5)(S_1,A_3)(S_2,A_{11})(S_3,A_1)(S_3,A_9)(S_3,A_7)$ $(S_3,A_{10})(S_4,A_8)(S_5,A_6)(S_5,A_{13})(S_5,A_{12})(S_6,A_2)$	2.16	2	0.22	3.33
$(S_1,A_1)(S_1,A_2)(S_1,A_3)(S_1,A_5)(S_2,A_6)(S_2,A_7)$ $(S_1,A_8)(S_3,A_9)(S_1,A_{10})(S_2,A_{11})(S_2,A_{12})(S_1,A_{13})$	4.33	2	0.5	3.33
$(S_1,A_8)(S_2,A_2)(S_2,A_{13})(S_2,A_3)(S_3,A_6)(S_4,A_9)$ $(S_5,A_{11})(S_6,A_{12})(S_7,A_1)(S_7,A_7)(S_7,A_5)(S_8,A_{10})$	1.6	2	0.16	3.25
$(S_1,A_7)(S_2,A_8)(S_3,A_1)(S_3,A_5)(S_3,A_{13})(S_4,A_9)$ $(S_4,A_{10})(S_4,A_3)(S_5,A_2)(S_6,A_{11})(S_7,A_6)(S_7,A_{12})$	1.8	2	0.23	3.28

Figure 4: Results of the second phase

5.3 Modeling of Phase 2

In this phase, we consider agility the most important goal of business. Thus regarding the connection between business goals and quality attributes, Granularity and Coupling should be considered. These features due to the high priority they have can help us in our choice. On the other hand, a series of other characteristics of user priorities (cost, time, availability, reliability) are considered in selecting services. We use IF-THEN rules to model user preferences. IF includes membership function of user verbal parameters and THEN includes membership function which is used to rank and score a particular concept. In fact, a fuzzy rule shows how a combination of user attributes is acceptable and at what rank. This approach can be a base for ranking the quality vectors of execution plans. More important rules are ranked higher. To create appropriate fuzzy rules, based on the variable attribute whose valued is desired less or more by the user, we should create a logical mapping between input and output variables. Fuzzy rules related to verbal variable of cost which is required in a less value by the user can be stated in the Figure (5).

CFcost 1. If (cost (dollar) is very cheap) then (Rank is very high)

CFcost 2. If (cost (dollar) is cheap) then (Rank is high)

CFcost 3. If (cost (dollar) is moderate) then (Rank is moderate)

CFcost 4. If (cost (dollar) is expensive) then (Rank is low)

CFcost 5. If (cost (dollar) is very expensive) then (Rank is very low)

Figure 5: The fuzzy logic variable of cost

However, for the variable of readiness which is required in a high value by the user, we use fuzzy logics in the Figure (6).

CFcost 1. If (cost (dollar) is very cheap) then (Rank is very high)

CFcost 2. If (cost (dollar) is cheap) then (Rank is high)

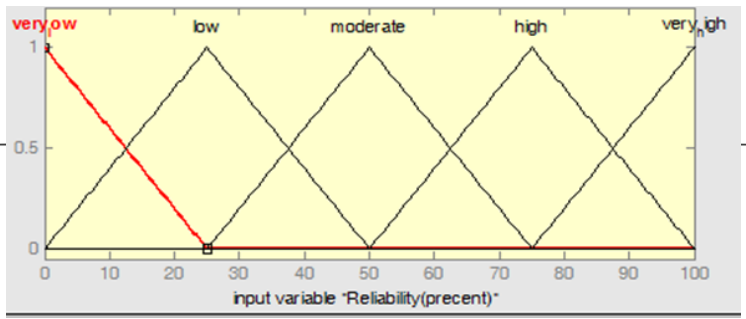
CFcost 3. If (cost (dollar) is moderate) then (Rank is moderate)

CFcost 4. If (cost (dollar) is expensive) then (Rank is low)

CFcost 5. If (cost (dollar) is very expensive) then (Rank is very low)

Figure 6: The fuzzy logic variable of availability

According to the mentioned descriptions for the system, we express (N=20) fuzzy logics which regarding the verbal variables of the system and their sections, the number of rules varies. The mentioned approach is implemented by MATLAB program and the corresponding membership functions are shown in Figure (7).



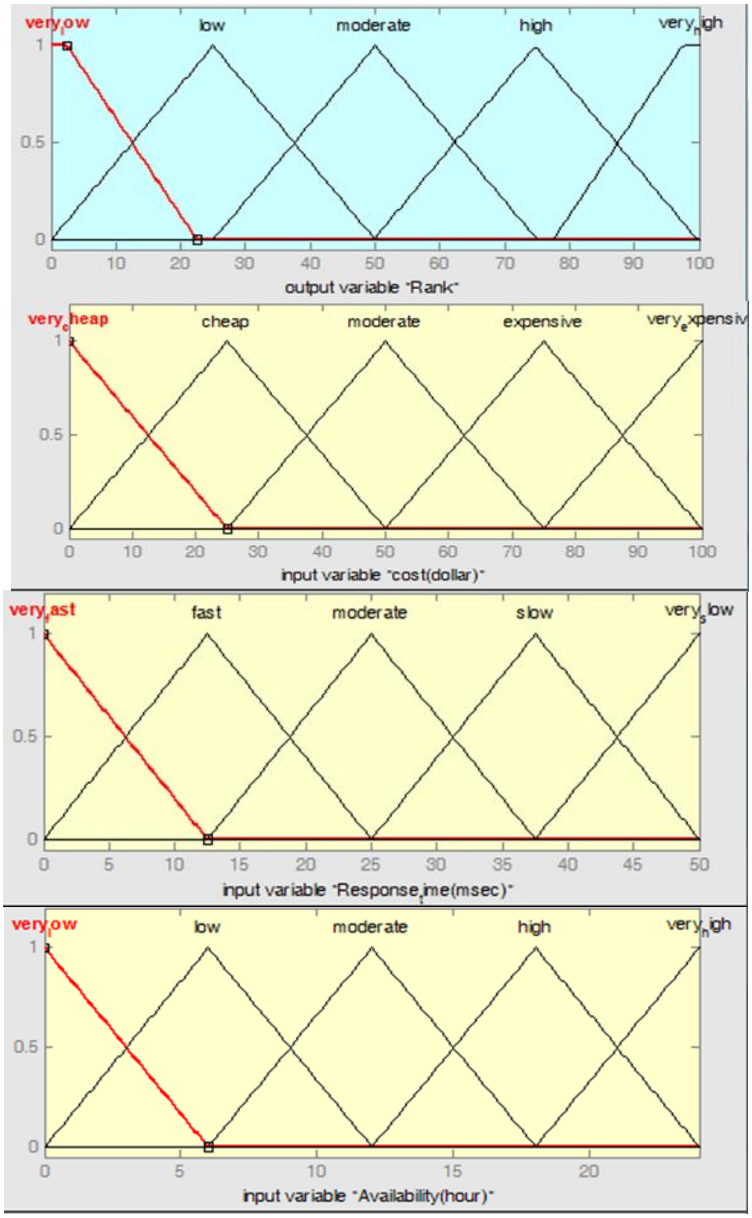


Figure 7: Membership functions of the system variables

However, for the variable of Response time which is required in a high value by the user, we use fuzzy logics in the Figure (8).

CFrt	11.If (Response_time (msec) is very_fast) then (Rank is very_high)
CFrt	12.If (Response_time (msec) is fast) then (Rank is high)
CFrt	13.If (Response_time (msec) is moderate) then (Rank is moderate)
CFrt	14.If (Response_time (msec) is slow) then (Rank is low)
CFrt	15.If (Response_time (msec) is very_slow) then (Rank is very_low)

Figure 8: The Logic Fuzzy variable of Response time

However, for the variable of Reliability which is required in a high value by the user, we use fuzzy logics in the Figure (9).

CFre	16.If (Reliability (precent) is very_low) then (Rank is very_low)
CFre	17.If (Reliability (precent) is low) then (Rank is low)
CFre	18.If (Reliability (precent) is moderate) then (Rank is moderate)
CFre	19.If (Reliability (precent) is high) then (Rank is high)
CFre	20.If (Reliability (precent) is very_high) then (Rank is very_high)

Figure 9: The Logic Fuzzy variable of Reliability

According to the state diagrams and the identified services, we assume there are two candidate services for A_6 and A_7 and there is one candidate service for other jobs and that the possibility of branching for A_6 and A_7 is equal to 0/5. The values of quality attributes are shown in Figure (10).

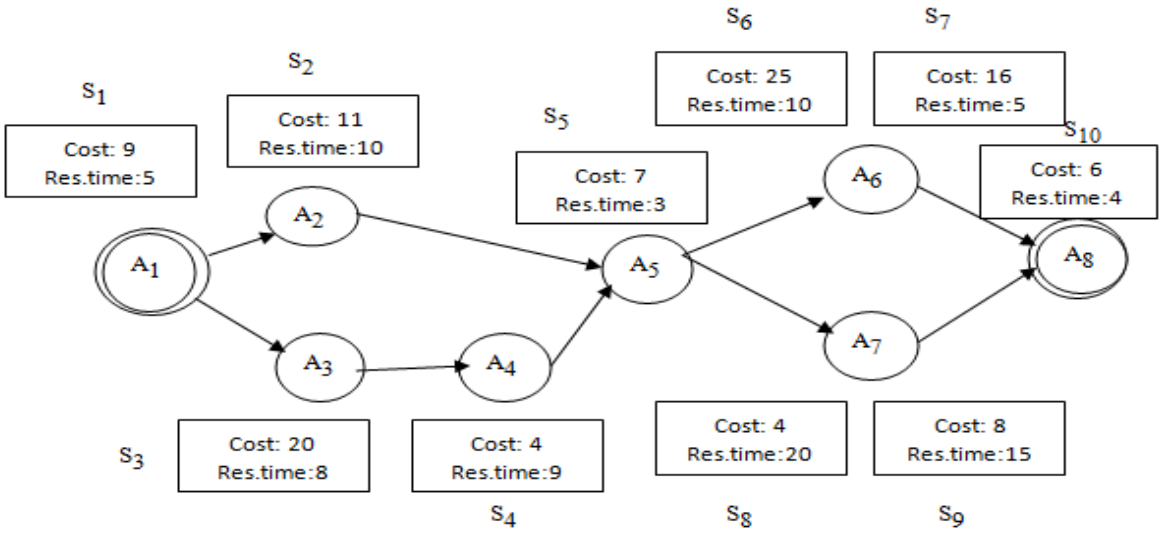


Figure 10: Services identified in the system

The calculated values of attribute cost and response time for the mentioned plans are determined using cumulative functions. The evaluated rate for each execution plan is stated in Table (3).

TABLE 3: ATTRIBUTE VALUES AND RATING OF EXECUTION PLANS

EX.Plan	$Q_{cost}(P)$	$Q_{res.t}(P)$	$Rank_1$	$Rank_2$
P_1	69.5	34	34.6	35.4
P_2	65	31.5	38.3	39.3
P_3	59	39	39.8	30.6
P_4	61	36.5	39.3	33.5

In the first case, with the assumption that the user gives 80 for the importance of cost criterion and 20 for the response time and zero for other criteria, the calculated rate for execution plan P3 using the implemented system is greater and thus it seems necessary to select this plan. But in the second case, if the user gives 20 for the importance of cost and 80 for the response time and zero for other criteria, the results lead us to choose execution plan P2. In spite of the quality vectors for an operation, we ask a number of users to determine the importance of each of the quality criteria and choose the best plan in their own ideas. Then we compare the results with the selected plan by the program. The comparison demonstrates the results which are induced by the mentioned approach correspond to that of users ideas. Even in those cases where the users were not determined in choosing from some quality vectors, the program could do it. As demonstrated, the users' ideas have a direct impact on the rate which is calculated for a composition of services. In fact, this is the user who by imposing his/her opinion chooses an appropriate composition in his/her idea.

6. Evaluation

The implementation is done in MATLAB, Assuming the availability of quality vectors for an operation. We want to determine the significance of each of qualitative criteria to select the best service. If you assuming value of 1 for factor of credit of fuzzy rules related qualitative criteria, Figure (11) shows a diagram of the system variables change and also change of variable costs, and thus it is rated for service composition.

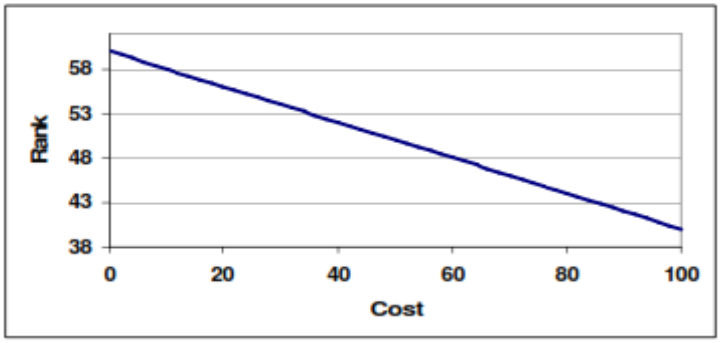


Figure 11: The effect variable cost on the rank

However, the reduction factor of credit rules associated with criteria and keep the rest is less the slope of diagram and width of each step in the ratings and this is also true for the rest of the criteria and reasons for the proper functioning of the system. Figure (12) shows the gap between the highest and lowest ranks rated services, or changes to the rules relating to qualitative criteria and keep the rest in the value of 1.

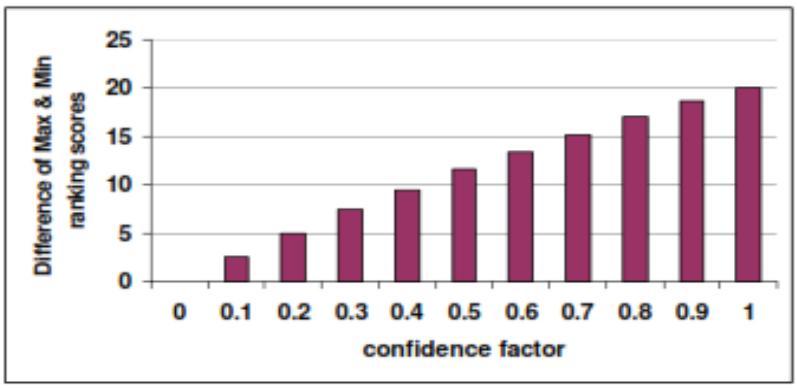


Figure 12: The effect of a factor on the width of Credit Scoring

However, user preference has direct impact at score is calculated for a composition of services. Thus, users choose right composition with their opinions.

Conclusion

In this paper, we have presented a method to identify the optimal composition of web services on genetic algorithms and fuzzy logic. Identifying services using GA result to identifying services aligned with business objectives and this lead to compatibility of business systems with business environment. In this paper, for the architect to make sure about the appropriateness of abstraction level, technical metric was used which can measure the attributes of service in designing levels. We also used GA to automate the identifying of services which contributed to an improvement in architect's role by automating tasks. To model the priorities of the user we used fuzzy rules with a logical mapping between input and output verbal variables. In this approach, after obtaining the degree of importance of each criterion from the users, based on the quality information of each execution plan, the composition with the highest rate is picked out. We can enumerate a couple of advantages for this approach. This approach emphasizes on the correspondence between the users ideas and the quality attributes of composition services. The user clearly expresses his/her priorities in choosing from the services with reviewing case study we must choice best plan with existing qualitative vectors and determine the importance each qualitative vectors. The user has the direct effort in the acceptable composition of services. Then this approach in web services selection and composition compare to previous approaches have upper efficiency and improvement.

References

[1] A. Rostampour, A. Kazemi, F. Shams, A. Zamiri and P. Jamshidi, "A Metric for Measuring the Degree of Entity-Centric Service Cohesion," Service oriented Computing and Application, 2010.

[2] F. Curbera, F. Leymann and R. Khalaf. "BPEL4WS", In IBM. http://www..106.ibm.com/developerworks/library/ws-bpelcol.html.

[3] G. B. Lamont, V. Veldhuizen and D. A. Coello, "Evolutionary Algorithms for Solving Multi-Objective Problems," Second Edition, Springer, 2007.

[4] I. B. Arpinar, B. Aleman-Meza, R. Zhang and A. Maduko, "Ontology-Driven Web Services Composition Platform", Proceedings of IEEE International Conference on E-Commerce Technology 2004.

[5] L. A. Zadeh, "Fuzzy Sets.", Information and control, 1965.

[6] L. Ackerman, A. D. Bari, G. Hodgkinson, A. Kesterton, L. Olson, B. Portier and U. Wahli, "Building SOA Solutions Using the Rational SDP," IBM Corp, 2007.

[7] L. G. Azevedo . "A Method for Service Identification from Business Process Models in a SOA Approach, " Springer, 2009, pp. 99–112.

[8] M. Josuttis. "SOA in Practice", First Edition. S.1: OReilly. 2007.

[9] M.ter Beek, A. Bucchiarone and S. Gnesi. "A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods". This work was partly supported by the EU project IST-3-016004-IP-09 Sensoria, 2009.

[10] M. Bell and E. A. Marks, "Executive's Guide to Service-Oriented Architecture. " Canada, John Wiley & Sons, New Jersey, 2006.

[11] N. Zhou, Y. Zhu, Y, H. Wang and Q. Ma, "Evaluating Service Identification with Design Metrics on Business Process Decomposition," International Conference on Services Computing, IEEE, 2009.
[12] R. Hull and J. Su,"Tools for Design of Composite Web Services", Proceedings of ACM SIGMOD 2004, 2004 June 13-18, Paris, France.
[13] R. Veryard and L. Wilkes, "Service-Oriented Architecture: Considerations for Agile Systems," http://msdn.microsoft.com/library/default.asp?url=/library/enus/ dnmaj/html/aj2service.asp, 2004.
[14] Randy L. Haupt and Sue Ellen Haupt. "Practical Genetic Algorithms", Second Edition, 2004.
[15] Sh. Dustdar and W. Schreiner. "A survey on web services composition.", Int. J. Web and Grid Services, 2005 Vol. 1, No. 1.
[16] S. Kumaran, "On the Duality of Information-Centric and Activity-Centric Models of Business Processes" CAiSE, Springer, 2008, pp. 32–47.
[17] S. Barbara, A. Balloni, F. Nogueira and F. Toda. "Information and Service-Oriented Architecture & Web Services", Proceedings of the International Conference on Enterprise Information Systems 2012, Portugal.
[18] V. Chifu. "Automatic Web Service Composition", In Technical University of Cluj-Napoca Faculty of Automatics and Computer Science PhD. Thesis, Cluj-Napoca, 2010.
[19] W.Zahreddine and Q.Mahmoud. "A Framework for Automatic and Dynamic Composition of Personalized Web Services", Proceedings of the 19th International Conference on Advanced Information Networking and Applications 2005.
[20] Y. Qing-mei, W. LAN and H. Dong-mei. "Fishery Web Service Composition Method Based on Ontology", Journal of Integrative Agriculture, 2012, vol. 11 no. 5, pp.792-799.

Authors

Hossein Momeni is an Assistant Professor of Software Engineering in Golestan University. He received the B.Sc. degree in software engineering from The Ferdowsi University of Mashhad, Mashhad, Iran and the M.Sc. and Ph.D. in software engineering from Iran University of Science and Technology (IUST), Tehran, Iran. His research interests include distributed system, real-time scheduling, wireless sensor actor networks and software engineering.

Fatemeh Mahdavi is a M. Sc. Student in Software Engineering in Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran. Her research interest is software engineering.