

SERVICE SELECTION BASED ON DOMINANT ROLE OF THE CHOREOGRAPHY

Dr. Ravi Shankar Pandey, Richa Pathak

Birla Institute Of Technology, Ranchi, India

ABSTRACT

Web services are playing dominant role on Internet for e-business. The compositions of these services are used to meet business objectives. The web service choreography describes the external observable behavior of these compositions. Many compositions may available for same functionality. These compositions cannot be distinguished on the basis of functional properties. This Quality of services (QoS) may help the user to select web services and to analyze composition of the web services. Web service choreography is going to dictate implementation of workflow. This workflow consists of several tasks. Each task is implemented by web services. These services are hosted in large numbers by different service providers on different service clusters. The mapping of service and task is difficult issue in run time environment. The interoperability between services is also a great problem. The selection of services is very big issue.

In this paper we have proposed a bio-inspired selection algorithm based on dominant role and proposed a discovery infrastructure. We have also used the client behavior to improve the failure of the composition of the service.

INDEX TERMS

Choreography, dominantRole, Web Service Selection

1 INTRODUCTION

Several business organizations are interacting their business activities through heterogeneous software applications. Many times they make interact within organization and some time with different organization. The software of different organizations may use different H/W resources and software platforms. The basic problem faced by industries is interoperability. The SOA service oriented architecture provides such solution. Software acts as a component or a service. These services are self describing, interoperable and loosely coupled. These services are hosted on different service registries. Any business activity may contain more than one sub activity and each activity may implemented by a one web service. To model the service interactions several standards are present like web service choreography and orchestration. Web service choreography defines common rules of interactions between several collaborating partners to perform a task. Collaborating partners may be web services or choreography. It consists of set of choreography, set of roles, relations between roles, activities to be performed, documents to be exchanged and reactions to the state of collaborating partners. Roles represent a subset of behaviour of web services. WS-CDL is an XML based language to describe web service choreography. This language is plate form and programming language independent. Web service choreography is used to model interaction between different organizations while orchestration is used to model interaction of one organization. In service oriented environment three actors are service provider, service requestor and service broker. Many service providers may publish similar services. In

such cases functional properties of services are not sufficient to distinguish among these services. In such cases non-functional requirements of user play deciding role with functional properties of the service in service selection. The semantic match is also important role in service selection to minimize failure of service composition. The present service descriptions do not captures this information in the WSDL file. In this scenario client invocation behaviour plays dominant role to minimize failure of service. Generally these services are stored in centralized environment. Due to tremendous growth in web services this centralized approach seems impractical. This motivated distributed environment to store web services. So many people prefer to store service using distributed environment. This environment may be peer to peer environment. Several research efforts have been made to index these web services. The peer to peer indexing uses Chord, CAN Pastry etc. approaches. The architecture of present P2P indexing system is like data lookup system and implement internet scale distributed has table. In peer to peer environment each peer has responsibility to search request of peer and if service is present then provide to the user or transfer query to neighbour peer and so on. In nature insects are using their wonderful sense to search food source. All they are using very good methods for interaction and communication in their society. All they are using path history to get the food. In insect society some insects are advertising the good path for all other insects. Like ant society uses pheromone to advertise good path and bees are selected good patches where good nectar may present. In this paper we have been proposed a bio-inspired algorithm to select best services. In this we also proposed a framework for service discovery which is also bio-inspired. We have also given the method for composition of web services using finite state machine. In this composition we have given the method to compute quality of service attributes values. So many quality of service attributes are proposed by different authors. We have only considered reliability, availability and latency. We have left network related quality of service attributes because they are neither in the control of service provider or the service requestor. We have used web service meta model with QoS as in [5] to use quality of service attributes in web services.

2. RELATED WORK

Pramodh N. et al.[13] have proposed a combined optimization and ranking mechanism for selecting web service. In their proposed mechanism optimization is done on the basis of ant pheromone deposition process and ranking is done on the basis of the QoS parameter evaluation. QoS parameters selected by the author are locality of reference, execution-time, access-count, availability. Locality of reference parameter is used to find out the most recently accessed service, access-count defines that how many times a service is fetched. Execution time is considered as the time taken by web service to provide service to the user. These parameters update themselves every time a service is executed. Changsheng Zhang et al have presented CASS algorithm for service selection. This algorithm uses clustering related shrinking process to provide direction of movement to the ants. Zongkai YANG et al[14] have presented a combination of Genetic algorithm and Ant Colony algorithm to compose web services in dynamic environment. Sunil R Dhore et al[15] have proposed a semantic composer that finds the optimal length among the all possible composition paths and recommend the best path. They have also proposed a framework using agents to provide negotiation during service composition. They have used divide and conquer methodology for parallel execution of composite web services. Alexander T et al[16] have presented a modified ant colony algorithm to evaluate QoS of web service. Web services are than selected to be composed using the result of the algorithm. Complete phenomenon is divided in to service selection phase, service processing phase, Graph building phase and ACO application phase to provide best output. They presented a bee-inspired optimization method which selects most optimal solution from web service composition. In this method they enhance the searching by using planning graph and matrix of semantic links which helps in getting optimal solution. They have also used QoS parameters and semantic matching of services to evaluate whether the solution is optimal or not. They proposed a method for optimizing to find

comprehensive quality for composite service. They have proposed a mathematical model with the help of QoS aggregation model and they have given an enhanced artificial bee colony algorithm (I-ABC). The I-ABC algorithm is based on taboo and chaos pattern and with the of result of some simulation experiment they proved feasibility and effectiveness of their model for web service composition. They have proposed approximation method which is based on QoS-aware service selection (SSP) and Artificial Bee Colony algorithm (ABC) in which they enhance ABC with greedy search algorithm. In this paper they have given an algorithm for obtaining neighbouring solution and the algorithm are pure online as canonical ABC. They have presented advance ABC technique for local search with more optimality and higher feasibility rate. In this paper they have proposed approximate-mapping and Van-Neumann strategy to get better local search algorithm and improved performance in searching.

3. OVERVIEW

Choreography and Social Insects:

Web service choreography is going to implement workflow of composite web services. This choreography has set of roles which implement a web service. One choreography may have more than one roles and one role may be used by more than one time in [2], called as DominantRole. This dominant role may play important role in service selection. In this paper, we propose a meta-heuristic search based on dominant role of choreography. Social insect's society works like a distributed system in which each member is very simple. Collectively they represent a complex social structure. Each member works in very discipline manner to meet the objective of their society. All individuals are self organized. They have excellent coordination among themselves. These insects have behaviours like foraging, division of labor, brood sorting, and cooperative transport, neighbourhood search (local search and global search) to provide food for their colonies. In this paper we propose the algorithm based on the searching mechanism of ant and bee.

Ant searching mechanism:

The most of ant species are blind. They are using chemical signature to move or search locations while humans and other species are using visual and acoustic mechanism. These chemical signatures are called pheromones. Some ant species are using trail Pheromone to make path in a ground for searching food. Some ant uses this trail path to search food. These two processes are called as trail laying and trail following phenomena. They are using indirect communication followed by pheromone deposited by other ant. Ant walking from nest to food deposited pheromone and formed a trail. Ant can smell these deposits to select the probabilistic path on the basis of strong smell.

Bee Searching Mechanism:

Bees can search their food up to a very long distance (14km) in multiple directions together to harvest nectar or pollen from the flower patches. A small number of bees constantly search flower patches. They are called scout bees. They are moving in every direction to search probability of food and harvested the food and deposited in hive. If any individual find high profitable food then go to the dance floor in hive and perform ritual dance called as **waggle dance**. Through this dance scout bees communicated the discovered flower patches to the idle bees; they started to exploit flower patches. The duration of the dance decides the rating of the scout food and more bees are recruited in harvesting of the nectar from the good rated flower

patches. Scout bees are working as discovery agent and as a advertiser. Recruiter may also perform waggle dance to recruit more idle bees to exploit flower patches.

Choreography and web services:

Choreography describes the interactions between services. It represents the workflow of the composite service. It consists of set of roles, relations and interactions between two roles. One Role may implement by more than one web services. An interaction takes place between to roles using the relationship. In this relationship one role acts as toRole and other role acts as fromRole. In any choreography more than one task may present and each task t_i may be implemented by web service w_{ij} where i is the task index and j is the index of web service.

Web Service and Quality of Service Attributes:

A. DAmbrogio [5] has proposed a meta model of web service, which include quality of services(QoS). They suggested reliability, availability and access control at service level, network related QoS attributes at port level, message related QoS at message level and operation latency and operation demand at operation level. In all these attributes we have used reliability, availability and operation latency as quality of service attributes. Network related attributes are neither in the control of service consumer and service provider. Further these attributes are based on other parameter like reliability is based on time between failure and expected failure; availability is based on expected availability, time to repair and expected failure. Operation latency is dependent on service time and turnaround time. Operation demand is computed from arrival rate of the operation requested by the client. Message QoS is computed from the message encryption quality. Network related QoS is derived from packet loss, bit rate and delay between two packets (jitter). These QoS have provided by service provider during hosting of the service. We have presented a tool which generates a WSDL file with QoS as given in [4].

Web services descriptions are stored in the form of web service description language (WSDL) file in the registry including functional properties (like input / output / preconditions) are stored. A. DAmbrogio [5] has proposed Q-WSDL meta model . We have also been also proposed to store WSDL details with QoS in the registry.

In service selection quality of service plays important role. We model set of services like node of a graph data structure. One role of choreography is going to implement any one service at one level of a graph and similarly other services are selected. These service selections make a path and decided by the non functional property of the service. As per my proposal this selection has been started form the dominantRole of the choreography and dominant Interaction as proposed [2]. The individual web service selection is dependent on quality of service values of attributes.

Bees Algorithm as in[18]:

1. Find Initial solution with random approach.
2. Apply fitness function to get solution form population
3. Form new population
4. Select area to find the neighbour
5. Determine the patch size and advertise it using waggle Dance.
6. Recruit bees for selected sites and apply fitness function.

7. 7Select advertiser bee from each patch
8. 8 Use rest bees for the same task.

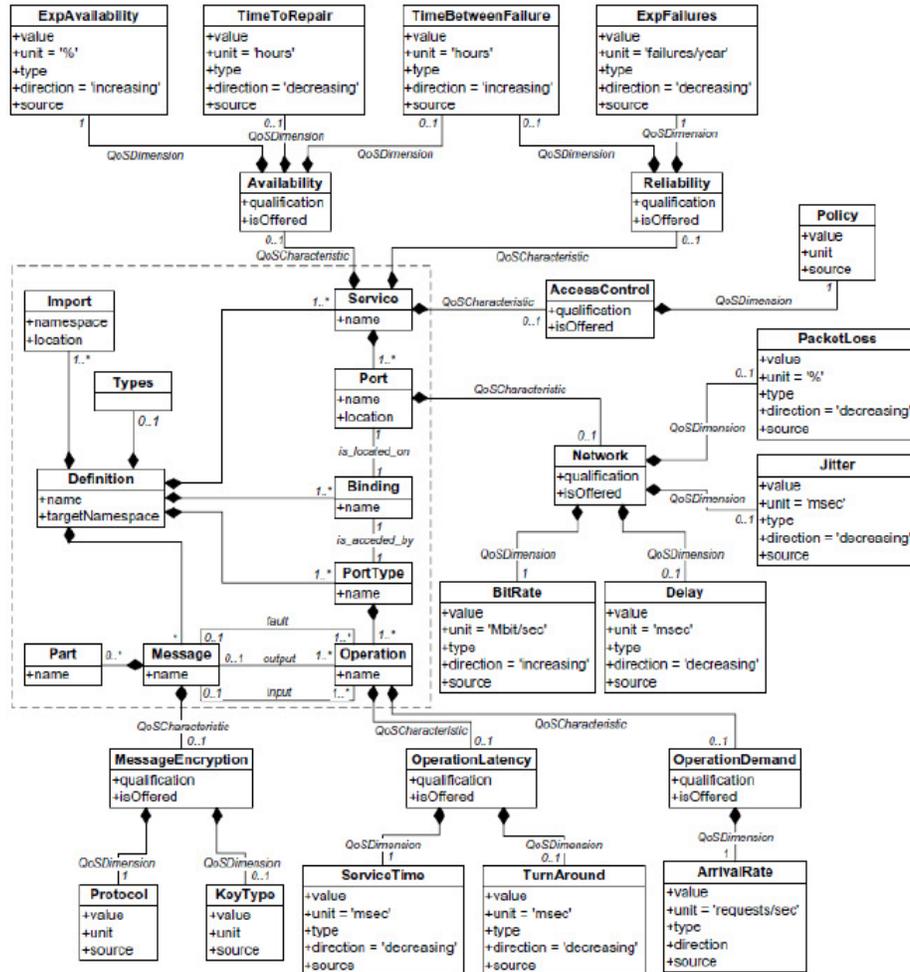


Figure 1 Meta Model of Web Service with QoS.

Ant Algorithm as in[15,16,17]:

1. Initialize solution randomly
2. Find the destination
3. Deposit the pheromone during return to the colony
4. Evaluation of path based on pheromone quality
5. Rest ants follow the same path

Proposed Methodology

Bee and ant searching algorithms are individually very unique and strong searching algorithms, here we decided to merge these algorithms to find the best optimal composition. In bee searching algorithm selecting highest optimal patch based on waggle dance number and in ant searching

algorithm colony pattern is the basic idea. The combination of both concepts has improved the searching mechanism. Our algorithm is inspired by both ant and bee concept, which is given as follows:

Algorithm Inspired by Ant & Bee:

1. Find Initial solution with random approach.
2. Apply fitness function to get solution form population
3. Form new population
4. Select area to find the neighbour
5. Determine the patch size and advertise it using waggle Dance.
6. Find the destination
7. Deposit the pheromone during travel from one node to another node.
8. Evaluation of path based on pheromone quality
9. Rest ants follow the same

4. EXTENDED SERVICE ORIENTED ARCHITECTURE

Service discovery architecture faces difficulty of interoperability between client request and dynamically discovered web services. WSDL files are used to match semantic details like input/output but do not contain details like invocation order of the operation.[6] proposed extended SOA architecture which resolves such kind of issues. They proposed interaction protocol service extension (IPSE) to capture the order of invocation of the client.

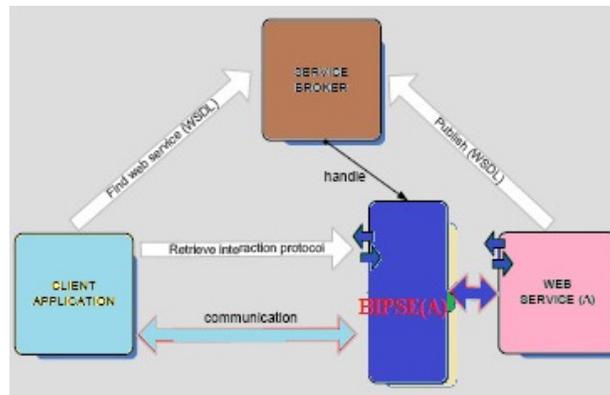


Figure 2 BIO-Inspired Extended Service Oriented Architecture.

They created this IPSE using client history. These IPSEs reduce the failure of web service invocation. Similarly we propose using these IPSE we construct protocol which tells about cluster with good IPSE services. We added one numeric number in each IPSE which records the number of the successful invocation of web services and in one cluster the summation of all such numbers represents the cluster usability. We have been decided the good patch (cluster) on the basis of these numbers. We have been proposed a new IPSE which includes successful invocation of web services and named as BIPSE (Bee inspired service extension protocol). Initially the BIPSE is created using WSDL file hosted by service provider on registry with invocation number as zero and order of invocation of operation as given in the

file. As client invokes service the order of invocation is refined over a time as given in[9] and number is updated by each successful invocation. We have been proposed a **waggledance protocol** which stores information about cluster quality. It takes number from all IPSEs present in same cluster and added if number is greater than the total number of web services in that cluster then **waggledance protocol** stores the difference between these numbers. The clusters are rated on the basis of **waggledance protocol and** good cluster has more waggledance number than others. [9]

Proposed Algorithm :

1. Cluster is selected on the basis of waggledance number
2. The first web service is selected on the basis of dominantRole of the web service choreography and quality of service attributes of the web services.
3. The next service selection is decided by the pheromone concept in ant colony method. Here pheromone means quality of service attributes.
- 3.1 If more than one path exists then which have more pheromone that path is selected.
4. Repeat step 3 to get the destination.

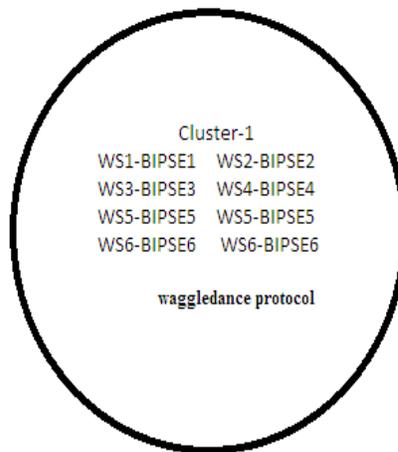


Figure 3 Clusters with Web Service BIPSE.

Bio Inspired Formal Model of Web Service with QoS Attributes:

We have modelled web service as a finite state machine which describes correct execution of web service. It consists of a tuple $(\Sigma, s_0, sf, \Delta, F)$

Σ : $\Sigma_A * \Sigma_q * \Sigma_{BIPSE}$ where Σ_A is set of activities, Σ_q is set of quality attributes of web services and Σ_{BIPSE} is store invocation order and number of successful invocations.

Σ_A (o1,o2...on) and Σ_q (rel,avl,c,lat) where oi is ith operation and rel, avl, c and lat are reliability, availability, cost and latency respectively. Σ_{BIPSE} (BIPSE-Bee inspired protocol service extension o1.o2.o3.on, n) where oi is the ith operation and chain represents order of invocation and n is the total number of successful invocation.

so is initial state and sf final state
 F: is set of final state

$\Delta: S * \Sigma \rightarrow S$ It takes set of input operations and quality of service attributes from one state and after transition transfers to other state. It returns a new state after accepting input operations, quality of service and bee inspired protocol service extension which client history of invocation order and number of successful invocation.

DominantRole:

It is a role of a web service choreography which appears maximum number of times in the interaction as proposed in [2]. Role captures the abstract behaviour of a web service. One role may implement many services which have same functionality. The dominantRole decides the source web service selection. In many cases it will be same as target web service defined in [8]. They are defined target web service is a service invoked by client.

Web Service Selection:

Waggledance protocol and DominantRole are decided the target cluster in which first web service is selected based on BIPSE. The next service selection is decided by the next web service invocation. So many web services may available for this invocation. This is decided by the quality of service attributes of individual web service provided by the service provider which is refined over a period of time. In this graph we are demonstrating the relationship between the services and their successful invocations. In the distributed environment these services are hosted in different registry domains.

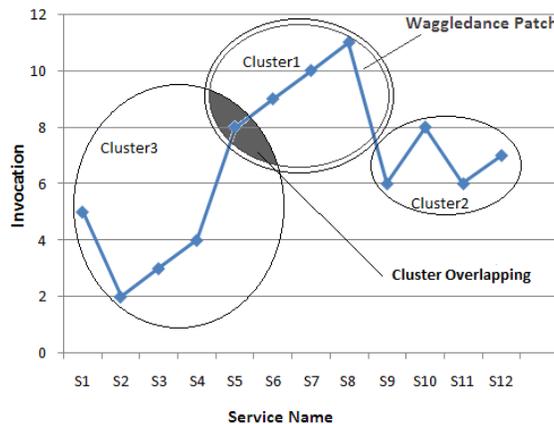


Figure 4 Graph between service, Invocation and their Clusters.

Several architectures [9, 11] have been proposed to organize these services in the form of cluster. Our proposal is based on the fact that history of successful invocation may help for the next time s service selection. As per our proposal we are showing in the good patches which are based on the waggle number as in the case of bees searching mechanism. As we have shown in the graph some services may overlap in the different cluster .In this graph double circle is representing the patch having highest waggle number.

Formal Model For Quality of Service of Web Service Composition:

We have considered reliability, availability and latency quality service attributes. Several research efforts have considered similar rule for service composition. We have considered in composition the reliability of composite web service is product of reliabilities of individual web service reliabilities. The availability of composite web service is also product of availabilities of individual web service availability. The latency of composite web service is summation of latency values of individual web services. The rules are summarizes as below.

Table 1 QoS Composition Rules.

QoS Attributes	Reliability	Availability	Latency	Cost
Sequence	Product	Product	Summation	Add
Parallel	Minimum	Minimum	Maximum	Add
Condition				
Loop	Same	Same	N*latency value	N*cost

We are giving rules of web service composition. The finite state machine consists of tuple , $F=(O,L,S,Q,\lambda, \alpha)$. Where O is set of operations, L is set of constrains, S is set of states, Q is set of quality of service attributes , λ is transition function from one state to another state and α is transition function which captures quality of service attributes. The composition rule is giving below.

$F1=(O1,L1,S1,Q1,\lambda1, \alpha1)$ and $F2=(O2,L2,S2,Q2,\lambda2, \alpha2)$ are two web services FSMs and resultant finite state machine is giving below.

$$F=(O,L,S,Q,\lambda, \alpha)$$

$$L=L1*L2$$

$$O=O1*O2$$

$$S=S1*S2$$

Given set of constraints $c=\{(O,i1\dots in): O \text{ is connected to } i1,\dots,in\}$

$\lambda =\{(A1,A2,S1,S2,B1,B2) \in \lambda \vee : O \text{ is connected to } (O,i1\dots in) \in (O \in B1 \cup B2 \text{ if and only if } ij \in A1 \cup A2 \text{ for all } j) \}$ Where A1 and A2 are subset of L1 and L2 respectively and B1 and B2 are subset of O1 and O2.

$$\alpha = \alpha1. \alpha2 = \{ql \in \alpha \text{ } qr=qr1*qr2 \text{ and } qa=qa1*qa2 \text{ and } ql=q11+q12 \}$$

$$\alpha = \alpha1 \parallel \alpha2 = \{ql \in \alpha \text{ } qr=\min(qr1,qr2) \text{ and } qa=\min(qa1,qa2) \text{ and } ql=q11+q12 \}$$

$$\alpha = \alpha1 \text{ loop } \alpha2 = \{ql \in \alpha \text{ } qr=qr1* \text{ or } *qr2 \text{ and } qa=qa1 \text{ or } qa2 \text{ and } ql=q11+q12 \}$$

In above expression qr,qa and ql are reliability, availability and latency quality of service attributes.

We have added the frequency of same invocation order and record the new invocation order with frequency one in all composition operators.

Role Model of Choreography:

In this model, we have recorded the quality of service attributes and invocation order with their frequency of usage by the users of a particular services. In this model one or more service may use same invocation order or one service may use different invocation order with different usage frequencies. In this model I have taken role name , Ai is used for role index, Li is used for set of variables, Oi is used for set of operations, Mi is used for set of messages and BIPSEi is used to record the invocation order for a service with service references and their usage frequencies .

RoleName = (Ai;Li;Oi;Mi;BIPSEi)

A model to discover web services using meta heuristic technique in distributed environments:

The following discovery architecture consists of clusters which are in Peer to peer environment. These clusters are stored web service registries. The registry contains information about web service functional as well as non-functional features using the web service description language with Quality of service attributes (WSDL-Q as given in [5]). This registry also contains client behaviour using bees concept which we called as BIPSE (Bee inspired service extension protocol). Each cluster contains waggledance protocol which advertises cluster rating as in the case of bees advertise the effective patches. They are updated using client feedback as ants are leaving pheromone during searching of food. The other ant follow same path using the pheromone deposited by the scout ant

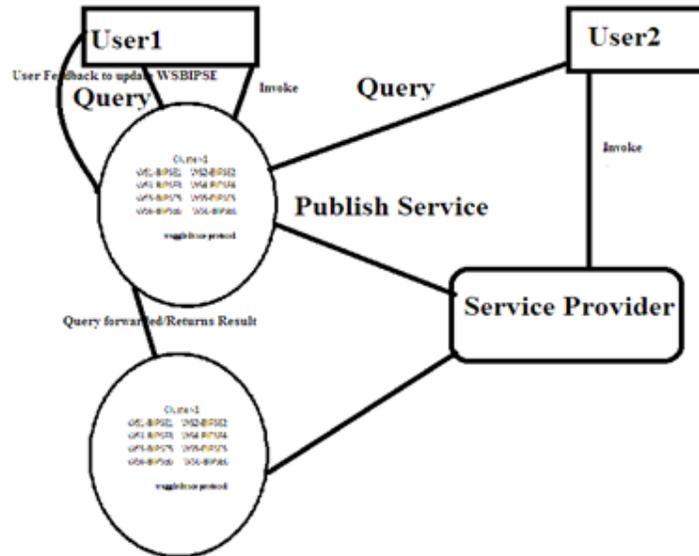


Figure 5: Registry Peer Structure:

In our case these Pheromone is successful invocation of web services and updated by the client in the registry

Registry peer contains the communication module, query processing module, match maker, QoS support module and service management module as given in[7]. The communication module acts as carrier of information and interacts with other registry peer, users, service provider and trusted domain. This module receives information regarding service advertisement and QoS data using WSDL-Q file provided by the service provider as in[5] and feedback using BIPSE protocol. It also provides interface to interact with other registry peers to forward query and exchange of service advertisement. The query processing module evaluate web service invocation matching based on input/output parameters and functionality of service and QoS range specified by the role as role QoS as in[3]. If the service QoS is with in range and their semantic details are matched by match maker module. This matching we have proposed on the basis of BIPSE(Bee inspired service extension protocol). The list of selected services is ranked on the basis of their QoS values and one service is selected on the basis of our proposed service selection algorithm.

Mapping of Web Service Based on Semantic Description & Quality of Service Advertisement to Peer Registries:

Example:

We are demonstrating our proposal using an example of travel plan. This travel plan consists of TravelAgent choreography which acts as root choreography and four other sub choreographies FlightReservation, TrainReservation, RoadTransport, Accommodation and Payment as given below.

```
TravelAgentChor =[ FlightReservationC ,
TrainReservationC , RoadTransportC , PaymentC ]
The details of individual choreographies are given below.
FlightReservationC=[queryFlight; reserveFlight ;
AccommodationC]
TrainReservationC=[queryTrain; reserveBirth;
AccommodationC ]
RoadTransportC=[queryBusqueryLimbo;
```

```
reserveBusu reserveLimbo;AccommodationC]
AccommodationC=[getAvailableAccF [] getAvailableAccT [] getAvailableAccR;
reserveRoomF [] reserveRoomT [] reserveRoomR;
confirmRoomF [] confirmRoomT [] confirmRoomR]
PaymentC=[makeP aymentV [] makePaymentM;
ackPaymentV [] ackPaymentV ]
```

In this example Train is dominantRole and Accommodation is dominantChoreography as proposed in[3]. These choreographies have roles TravelAgent, Train, Hotel, Bus, AirBus and Limbo. Each role contains role name, set of operations, set of message, set of web services successfully implemented this roe and quality of service attributes. We have considered only reliability, availability and latency QoS. We have proposed method for QoS computation of a role based on QoS of competing web service in[3]. We have considered QoS as range in[3]. Similarly we considered web services have a name, set of operations, set of messages and quality of service attributes of the web service like latency, reliability and availability.

QoS of Role References:

```
C1 := Customer1(O1,BIPSE1)
O1:= {getT ravelItinerary1,queryFlight1,queryTrain1
```

```
,queryRoad1,queryPayment1 }
OQ1:= {20;70;40;50;10}
Rel1= (150;200)
Av11= (100;400)
BIPSE1=<gT,qF ,qT,qR,qP,20>
```

T1:= TravelAgent(O1, BIPSE1)

This stores the order of invocation of operations and number of successful invocation. In this 10 is number of successful invocation. Initially the order of invocation of operations is same as in the WSDL file but it refines over a period of time.

```
BIPSE1=<qF,sF ,sR,qT,sTqR,gT,rP,cP,10>
```

```
O1={queryF1, selectF1 , selectR1,queryT1,selectT1 ,queryR1,getTravelI1
,requestP1,;confirmP1 }
```

```
OQ1:= {20;30;40;10;40;60;20;50;30}
```

```
Rel1= (100;300)
```

```
Av11:= (100;300)
```

```
C2 := Customer1(O2,BIPSE2)
```

```
O2:= { getT ravelItinerary1,queryFlight1,queryTrain1
,queryRoad1,queryPayment1 }
```

```
OQ2:= {20;70;40;50;10}
```

```
Rel2= (150;200)
```

```
Av12= (100;400)
```

```
BIPSE2=<gT,qP ,qT,qR,qF,30>
```

T2:= TravelAgent(O2,BIPSE2)

This stores the order of invocation of operations and number of successful invocation. In this 20 is number of successful invocation. Initially the order of invocation of operations is same as in the WSDL file but it refines over a period of time.

```
BIPSE2=<sR,qT, qF,sF ,sT,qR,gT,rP,cP,20>
```

```
O2={queryF1, selectF1 , selectR1,queryT1,selectT1 ,queryR1,getTravelI1
,requestP1,;confirmP1 }
```

```
OQ2:= {20;30;40;10;40;60;20;50;30}
```

```
Rel2= (100;300)
```

```
Av12:= (100;300)
```

A1= AirLine(O1, BIPSE1)

```
O1 = {queryF1 ; selectF 1 ;confirmF1 ;queryH1 ;reserveR1; cofirmR1 }
```

```
OQ1 = {20;30;40;50;30;40}
```

```
Rel1= (100;300)
```

```
Av11= (100;300)
```

```
BIPSE1={qF ; sF ;cF ;qH ;rR; cR,20}
```

H1= Hotel(V1,M1,O1, BIPSE1)

```
O1= {queryR1,reserveR1,confirmR1 }
```

```
OQ1={20;30;10}
```

```
Rel1= (150;400)
```

```
Av11= (200;600)
```

```
BIPSE1={qR,rR,cR,30}
```

Tr1 = Train(V1 ;M1 ;O1 , BIPSE1)

O1= {queryT1 ;reserveT1 ;queryH1 ;reserveR1;confirmR1}
 OQ1= {40;50;10;20;30}
 Rel1= (100;500)
 Av11= (200;500)
 BIPSE1={qT ;rT ;qH ;rR;cR,50}

B1 = Bus(V 1 ;M1 ;O1, BIPSE1)
 O1 = {queryB1 ;reserveB1 ;queryH1 ;reserveR1;confirmR1}
 OQ1 = {40;50;30;80;90}
 Rel1= (500;700)
 Av11= (200;600)
 BIPSE1={qB ;rB ;qH ;rR;cR,60}

L1 = Limbo(V1,M1,O1, BIPSE1)
 O1= {queryL1 ;reserveL1;confirmL1;queryH1 ;reserveR 1; confirmR1}
 OQ1 = {20;30;40;10;40;60}
 Rel1= (100;300)
 Av11= (100;400)
 BIPSE1== {qL ;rL;cL;qH ;rR; cR,40}

M1=Master(O1, BIPSE1)
 O1 = {requestP1 ;paymentA1 }
 OQ1 = {10;20}
 Rel1= (250;300)
 Av11= (200;700)
 BIPSE1={rP ;pA ,30}

V1=Visa(O1, BIPSE1)
 O1 = {requestP1 ;paymentA1 }
 OQ1 = {10;20}
 Rel1= (250;300)
 Av11= (200;700)
 BIPSE1={rP ;pA ,30}

Role QoS

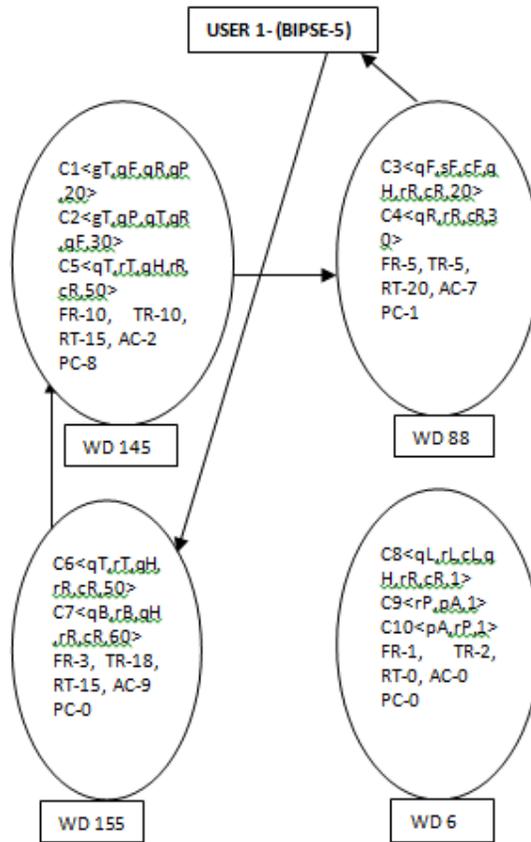
Customer1 = (A1;O1;BIPSE1 ,BIPSE2)
 O1 = {getTavellitinerary;queryFlight;queryTrain; queryRoad; queryPayment}
 OQ2 = {< 20; 40 >;< 70; 20 >;< 40; 30 >;< 50; 20 >;< 10;10 >}
 RF1 = {C1;C2}
 BIPSE={ BIPSE1 ,BIPSE2}
 BIPSE={ <C1,<gT,qF ,qT,qR,qP,20>>,<C2, <gT,qP ,qT,qR,qF,30>>}

TravelAgent = (A2;O2;BIPSE)
 O2 = {queryF; selectF;queryT; selectT; queryR; selectR;getT ravell;requestP;confirm}
 OQ2 = {< 10; 20 >;< 30; 35 >;< 20; 40 >;< 10; 20 >;< 40;60 >;< 30;60 >;< 20;50 >;< 50;60 >;< 10;30 >}
 RF2 = {T1;T2}
 Rel2 := (100;300)
 Av12 := (100;300)
 RQ2 := (Rel2;Av12;OQ2)

BIPSE={BIPSE1,BIPSE2}
 BIPSE={<T1, <qF,sF ,sR,qT,sTqR,gT,rP,cP,10>
 >,<T2, <sR,qT, qF,sF ,sT,qR,gT,rP,cP,20>>}
 AirLine3 = (A3;L3;O3;M3;RF3)
 O3 = {queryF;reserveF;cofirmF;queryH;reserveR; cofirmR}
 OQ3 = {< 20; 20 >;< 30; 59 >;< 10; 23 >;< 30; 50 >;<
 30;40 >;< 40;50 >}
 RF3 = {A1;A2}
 Rel3 = (100;400)
 Avl3 = (100;470)
 RQ3 := (Rel3;Avl3;OQ3)
 BIPSE={<A1, <qF,sF ,sR,qT,sTqR,gT,rP,cP,10>
 >,<A2, <sR,qT, qF,sF ,sT,qR,gT,rP,cP,20>>}
 Hotel4 = (A4;L4;O4;M4;BIPSE1,BIPSE2)
 OQ4 = {< 20;30 >;< 30;40 >;< 10;20 >}
 RF4 = {H1;H2}
 Rel4 = (100;400)
 Avl4 = (100;470)
 RQ4 := (Rel4;Avl4;OQ4)
 BIPSE={<H1,H2> {qR;rR;cR,30}>}
 Train5 = (A5;L5;O5;M5;BIPSE1,BIPSE2)
 O5 = {queryT;reserveT;queryH;reserveR;cofirmR}
 OQ5 = {< 40; 50 >;< 30; 50 >;< 10; 40 >;< 20; 60 >;<30;70 >}
 RF5 = {Tr1;Tr2}
 Rel5 = (100;500)
 Avl5 = (200;770)
 RQ5 := (Rel5;Avl5;OQ5)
 BIPSE={<Tr1, {qT ;rT ;qH ;rR;cR,50}>,<Tr2, { qH ;rR;cR,qT ;rT ; 40}>}
 Bus6 = (A6;L6;O6;M6;BIPSE1,BIPSE2)
 O6 = {queryB;reserveB;queryH;reserveR;cofirmR}
 OQ6 = {< 30; 40 >;< 50; 50 >;< 40; 50 >;< 10; 30 >;<50;80 >;< 40;90 >}
 RF6 = {B1;B2}
 Rel6 = (200;700)
 Avl6 = (200;600)
 RQ6 := (Rel6;Avl6;OQ6)
 BIPSE={<B1, {qB ;rB ;qH ;rR;cR,60}>,<B2, { qH ;rR;cR, qB ;rB ;40}>}

 Limbo7 = (A7;L7;O7;M7;BIPSE1,BIPSE2)
 O7 = {queryL;reserveL;queryH;reserveR;cofirmR}
 OQ7 = {< 20; 20 >;< 30; 30 >;< 40; 50 >;< 10; 20 >;<
 30;40 >;< 40;50 >}
 RF7 = {L1;L2}
 Rel7 = (100;300)
 Avl7 = (100;400)
 RQ7 := (Rel7;Avl7;OQ7)
 BIPSE={<L1, {qL ;rL;cL;qH ;rR; cR,40}>,<L2, { qH ;rR; cR, qL ;rL;cL;60}>}
 V isa8 = (A8;L8;O8;M8;BIPSE)
 O8 = {requestP;payment}
 OQ8 = {< 10;20 >;< 20;30 >}
 RF8 = {V 1;V 2}
 Rel8 = (100;300)
 Avl8 = (100;400)
 RQ8 = (Rel8;Avl8;OQ8)

BIPSE={<V1,V2 {rP ;pA ,30}>
 Master9 = (A9;L9;O9;M9;BIPSE)
 O9 = {requestP;payment}
 OQ9 = {< 10;50 >;< 20;70 >}
 RF9 = {M1;M2}
 Rel9 = (150;300)
 Av19 = (100;700)
 RQ9 := (Rel9;Av19;OQ9)
 BIPSE={<M1,M2 {rP ;pA ,100}>}



In this example we have taken different web services for different roles. We are using 5 roles to demonstrate our example with 10 different web services. In our model 'wi' indicate web service name, 'r name' indicate role name, 'opr' for set of operation, 'BIPSE i' Bee inspired service extension protocol where 'i' is for different web service (i = 1.....no. of web services). We have taken 7 web service choreographies related to travel domain, Travel agents (TA), Train reservation (TR), Flight Reservation (FR), Road Transport (RT), Hotel Reservation (HR), Accommodation (AC), and Payment (PC). Each choreography is a group of one or more web services. web service qT(query train), rB(reserve birth) are in TR, qF(query flight), rF(reserve flight) are in FR, qB(query bus), qL(query limbo), rB(reserve bus), rL(reserve limbo) are in RT, gAF(get availability flight), Here we have taken 4 clusters, each cluster consist of different web

services of similar domain. Cluster 1 is having waggle dance (WD) number 145 as counted by total number of successful invocation of services within cluster 1. Waggle dance number of cluster 2, 3 and 4 is 88, 155, and 6 respectively.

CONCLUSION

In this paper we have proposed bio-inspired service selection algorithm for a given web service choreography. In this selection algorithm we have proposed new search algorithm. This is based on natural selection. We have used ant and bees searching algorithm together. We have used bees algorithm for finding good (patches) clusters.. Ant algorithm is used for finding good services within a cluster on the basis of their QoS parameters as ant user for finding their paths using the pheromones for the service selection. In this strategy we are using dominant role concept of the choreography to start our service selection.

We try to find new service selection techniques which will include new QoS parameters like Geographical Affinity and also based on usage behaviour of the web services.

REFERENCES

- [1] Ravi Shankar Pandey and B. D. Chaudhary, "A cost model for participating roles based on choreography semantics " IEEE, 2008 Asia-Pacific Services Computing Conference(APSCC08), pp.277--283 , 9--12 Dec. 2008, Tiwan , doi: 10.1109/APSCC.2008.117
- [2] Ravi Shankar Pandey and B. D. Chaudhary, " A Meta-Model Based Proposal for QOS of WSDL choreography " in Proc. Int. Multi Conf. of Engineers and Computer Scientists (IMECS-10) pp. 677-683 , 17--19 March. 2010, Hong --Kong , ISBN: 978-988-17012-8-2
- [3] Ravi Shankar Pandey and B. D. Chaudhary, " An Estimation of Min-Max of QoS Attributes of a Choreography" in Proc. Advances in Engineering, Science and Management (ICAESM12) pp.872--878 , 30-31 March 2012, Nagapattinam, Tamil Nadu ,ISBN: 978-1-4673-0213-5
- [4] Ravi Shankar Pandey " A Platform For Service Provider To Advertise Quality of Service Using WSDL " International Journal of Information(Acquisition (IGIA)of Word Scientific) accepted for publication.
- [5] A. DAmbrogio," A model-driven wsdl extension for describing the qos of web services", In Proc. of IEEE Int. Conf. on Web Services(ICWS 06), Washington, DC, USA,2006,p.p. 789-796
- [6] G. Denaro, M. Pezz' e, D. Tosi and Daniela Schilling, "Towards Self-Adaptive Service-Oriented Architectures", In Proc. of 2006 workshop on Testing, analysis, and verification of web services and applications, July 17, 2006, Portland, Maine, USA, ACM , p.p. 10--16
- [7] Le-Hung Vu, Manfred Hauswirth and Karl Aberer," Towards P2P-based Semantic Web Service Discovery with QoS Support", (Data, Information, and Process Integration with Semantic Web Services, pp.1--15
- [8] San-Yih Hwang, Wen-Po Liao, Chien-Hsiang Lee," Web Services Selection in Support of Reliable Web Service Choreography", In Proc. of 2010 IEEE International Conference on Web Services, p.p. 115--122
- [9] Mohamed Sellami, Samir Tata, Zakaria Maamar and Bruno Defude, "A Recommender System for Web Services Discovery in a Distributed Registry Environment," In: 4th International Conference on Internet and Web Applications and Services (ICIW'09), IEEE May 2009, pp. 418-423.
- [10] Natallia Kokash, Aliaksandr Birukou and Vincenzo D'Andrea, "Web Service Discovery Based on Past User Experience," In: 10th International Conference on Business Information Systems (BIS'07), Springer April 2007, vol 4439, pp 95-107.
- [11] Freddy Lecue, "Clustering Social Networks," In: 5th International Workshop (WAW'07), Springer December 2007, p.p 57-67.
- [12] A. Birukou, E. Blanzieri, P. Giorgini, and N. Kokash, "Improving web service discovery with usage data," In: Software IEEE December 2007, vol 24, issue no. 6, p 47-54

- [13] Pramodh N , Srinath V , Sri Krishna A ,” Optimization and Ranking in Web Service Composition using Performance Index”, Vol 4 No 4 Aug-Sep 2012, International Journal of Engineering and Technology (IJET),pp-208-213.
- [14] Zongkai YANG, Chaowang SHANG†, Qingtang LIU, Chengling ZHAO,” A Dynamic Web Services Composition Algorithm Based on the Combination of Ant Colony Algorithm and Genetic Algorithm” Journal of Computational Information Systems , vol 6, Aug-2010 , pp-2617-2622.
- [15] Sunil R Dhore, Prof. Dr M U Kharat ,” QoS Based Web Services Composition using Ant Colony Optimization: Mobile Agent Approach”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 7, September 2012, pp-520-527.
- [16] Alexander T ,E. Kirubakaran, “Optimal QoS based Web Service Choreography using Ant Colony Optimization”, International Journal of Computer Applications ,Volume 102 – No.11, September 2014, pp-39-46.
- [17] Changsheng Zhang, Hao Yin, and Bin Zhang , “A Novel Ant Colony Optimization Algorithm for Large Scale QoS-Based Service Selection Problem”. Discrete Dynamics in Nature and Society, Volume 2013, 2013, Article ID 815193, pp 9.
- [18] R. Sagayam, K. Aklidansewari, “Comparison of Ant Colony and Bee Colony Optimization for Spam Host Detection,” In: International Journal of Engineering Research and Development (IJERD’08), November 2008, vol 4, issue 8, pp 26-32.