AN ADAPTIVE APPROACH FOR DYNAMIC RECOVERY DECISIONS IN WEB SERVICE COMPOSITION USING SPACE BASED QOS FACTOR

Dr. Ravi Shankar Pandey, Richa Pathak

Department of Computer Engineering, BIT Mesra , Ranchi, India

ABSTRACT

Service Oriented Architecture facilitates automatic execution and composition of web services in distributed environment. This service composition in the heterogeneous environment may suffer from various kinds of service failures. These failures interrupt the execution of composite web services and lead towards complete system failure. The dynamic recovery decisions of the failed services are dependent on non-functional attributes of the services. In the recent years, various methodologies have been presented to provide recovery decisions based on time related QoS (Quality of Service) factors. These QoS attributes can be categorized further. Our paper categorized these attributes as space and time. In this paper, we have proposed an affinity model to quantify the location affinity for composition of web services. Furthermore, we have also suggested a replication mechanism and algorithm for taking recovery decisions based on time and space based QoS parameters and usage pattern of the services by the user.

KEYWORDS

Failure recovery, QoS Factors, Location Affinity, Recovery Decisions

1. INTRODUCTION

Enterprises are using information technology to automate their business activities within the organization as well as with other organizations. They are using different software and hardware platforms for automating these activities. In general these business activities consist of more than one smaller sub activities. These activities may reside at same geographical location or at different geographical location. Internet provides an infrastructure for integrating such activities. A new paradigm is developed to facilitate these integrations known as Service Oriented Architecture (SOA). In this architecture software's are viewed as services. This architecture contains description of the service using WSDL (Web Service Description Language), communication protocol SOAP (Simple Object Access Protocol) for providing interaction among the services and UDDI (Universal Description, Discovery and Integration) storage mechanism for these services. All they are based on open source mechanism provided by XML (Extensible Markup Language). These services are loosely coupled in nature and may be integrated at run time dynamically. This architecture facilitates three types of entities; these are service requester, service provider and service broker. The main role of the service providers are to host their services in storage area(UDDI) and requester searches the desired service through broker in automated environment.

Different service providers may publish the same services in the UDDI registry. So it become strenuous to select a single service from the same service domain based on their functional features. In these view non-functional properties of the services play an important role in service selection. These non functional properties are known as QoS attributes. The service providers consider these parameters as QoS attribute for their published services while service requester uses these parameters as non-functional requirement for their desired services. These parameters are availability, execution time, cost, reliability, latency. All these parameters are primarily based on time factor. .[24] have proposed QoS attributes which are based o the usability of the services in their composition other than time related QoS (dominant role, dominant operations).Web services are hosted at different geographical locations and are used by people situated at different geographical regions. In this scenario services may be preferred by their geographical location or they may be frequently used for a particular region. Thus we have concluded that for service selection location is also a non-functional attribute of a service. We are categorizing all the non-functional attributes or features of web service in two categories. One is the QoS based on time factor (reliability, cost, latency, execution time) and other is the QoS based on location that is location affinity of the service. We have introduced a term “location affinity” with respect to the web services as a space based quality of service factor. We are describing the location affinity as the degree or extent to which user likes and utilize the service of a particular geographical location. When a service execution takes place in atomic manner it is less susceptible to failure. Service composition is prone to failure due to lack of availability, semantic mismatching and other functional attributes. Several research efforts have been made to model non-functional parameters and provide a recovery mechanism in case of the failure occurred during compositions.

In this paper, we have proposed a methodology for taking the recovery decisions to handle the failure in service composition based on time as well as location based QoS attributes together. We are also proposing finite state machine model for the web service composition. This model includes their QoS parameters, input parameters, set of operations, transition function for transition from one state to another and evaluating impact on Overall QoS factors of our model. We are also suggesting a methodology to take recovery decisions dynamically.

This paper is organized as follows. The section 2, introduces the related work on service composition, service failure, QoS evaluation using finite state machines. In section 3, we represented the location affinity based meta-model and finite state machine model for service composition using QoS attributes. In the last section, we presented a replication strategy and algorithm to take recovery decision in the presence of fault.

2. Related Work

Now a day’s web based services are being capturing the IT market completely. With the growth of use of Internet user’s requirement is also growing explosively and single service fails to meet this requirement. Thus service composition came in to the existence. Various methods for service composition have been presented whether it is dynamic or static. Sunil R Dhore et al [1] have discussed the semantic composer that uses enhanced ant colony optimization mechanism to provide service composition in efficient manner by finding optimal composition length in each set of candidate web service proposed for service composition. Zhou Xiangbing et al [2] have presented a web service modeling ontology for service composition using genetic algorithm. In this paper they have considered four non functional attributes, these are: information exchange, time, matching reasoning and cost. Finite state machine based composition solutions have also been proposed by different authors.Olga et al [4] have estimated the quality of service and quality of experience of web service using the finite state machine concept. Jun Sun et al [25] have suggested the finite state machine synthesis for software systems. Number of methods has been proposed to monitor and identify faults in service based systems. Jocelyn Simmonds et al [9] have presented an approach to monitor and recover web service based applications using three stage phenomenons. In first stage of preprocessing BPEL code is converted in to labeled transition systems and properties supplied by the user are converted in to monitor code .In monitor phase application is checked for an error and In recovery phase compensation plan are ranked and applied. Compensation allows a web service to go-back to the previous stage if an error occurred. Thirumaran.M et al [6] have proposed a QoS based run time exception handler. This handler first takes exception information as an input and on the basis of that calculates the time and space complexity. Decidability evaluator than categorize the problem in to, NP-Complete solvable problem set, NP-Hard may be solvable problem set. They have also used turning machine to find the solution of the problem in a specified time using audit log that contains some predefined solutions for particular problem. Services composed at run time are more prone to failure. These failures could occur due to various reasons .Hadi Saboohi et al [11] have figured out reasons for the failures. This paper identifies the functional as well as non functional causes of service composition failure. Functional causes includes service unavailability, service malfunctioning and non functional causes includes unexpected data, network delay, response time-out. Abdelkarem Erradi et al [17] have proposed various recovery policies (retry, substitution, parallel execution, dynamic update of service composition). They have also proposed MASC (Manageable & Adaptive Service Composition) model for monitoring failure and providing recovery from failure. H. Elfawal Mansour et al [14] have proposed a model that calculates the reliability at run time and uses roll back mechanism to recover services from failure. They have used a broker that decides that the result computed by a particular web service is relevant or not. If it is relevant process of computation continues otherwise the service is roll backed and another service having same functionality is been called to complete the execution of the complete web service based system. Suchi Gupta et al [8] have presented subset replacement mechanism to handle service failure. According to this paper failed set of services are first identified by the middle agent and then replaced by another set of services providing the similar functionality. Keting Yin [15] have also used replacement mechanism to recover service from failure. They have also presented a ranking mechanism to select the service that could replace the failed service based on non functional QoS parameters. Guisheng Fan et al[16] has categorized the service failure as failure of available service, failure of component service, Failure of operating environment and model the composite web services using petri-nets. Cao et al [18] have presented in which firstly, the service execution graph (SEG) is introduced and a service execution solution selection algorithm is proposed to choose one from the solution set of TSSA which has the highest success rate of recovery. Then, the concepts of execution backup path and switch cost are introduced and a search algorithm is presented to search for the optimal backup path when current service failure occurs.Rafael Angarita et al [10] have categorized the recovery mechanism in to two parts based on the ACID properties of a transaction. These categories are forward and backward recovery mechanism. In forward recovery mechanism replacement, compensation and replication strategies are included. In backward recovery mechanism roll-back strategy is included. All the recovery mechanism has been tested based on the execution cost and execution time in different failure conditions and dynamic recovery decisions could be taken on the basis of experimental result. Replication of the web service is the preferable solution of handling failure of the services in distributed environment. Marwa F. Mohamed et al [21] have provided a framework for the dynamic replication of the web services .This framework automatically replicates a service based on it's consumption and reduces the service response time as well. They have taken sensor based mechanism to identify the service failure and then take replication decision for the failed web service. Mario Bravetti et al [22] computed the system performance after replicating the web service using SOCK calculus. An Liu et al [26] have proposed a framework named as "FACTS" to provide highly reliable and fault tolerant service composition. They have also used BPEL to elaborate their proposed architecture.Johannes Behl et al [27] have presented architecture to replicate the BPEL engine as well as web service using proxy servers (input and output proxy servers) to enhance the availability and reliability of the services in cloud computing platform. Ivona Brandic et al [23] location affinity QoS with respect to the Grid Computing environment. It is basically used for providing the security to the users of the grid and giving the grid resources of particular domain (organization, home) based on the user's willingness. R.S.Pandey et al [24] has proposed a methodology to estimate the minimum and maximum value of the Quality of service attributes (reliability, availability, latency, security). We have analyzed that all the above researches basically emphasis upon the time based Quality of Service factors for composing the service and also for recovery decision. In [10] recovery decisions were taken dynamically on the basis of the execution time and execution cost. None of the author considered the geographical location factor to take recovery decisions at run time. In our paper we are selecting recovery strategy among all the available strategies (replacement, roll-back, replacement) based on geographical location .In this paper we are also advocating the need of the replication mechanism and suggested a methodology to decide which service from the same location affinity and similar set of functionality should be replicate in distributed environment to avoid failure. In our proposed work we are including location factor for the composition of services based on finite state machine. Dynamic recovery concept has also been extended with location necessity.

3. MODELING LOCATION AFFINITY

In [23] location affinity is modeled to benefit the grid computing users. The authors of this paper mainly concerned with security and legal issues, while using grid technology. They have also discussed that grid users hesitate to use this technology everywhere rather they wanted to use this technology within a specific domain or range. They have modeled location affinity for grid user with in a specific domain of their choice for the security purpose. QoWL (QoS-aware Grid Flow Language) that is a XML based language is developed and used for modeling the location affinity. In [23] location affinity is modeled for the security purposes but in our paper we are modeling location affinity with a different perspective. We are considering affinity as the user's preference to use a particular service that belongs to a certain geographical boundary. For example a user weekly visits to the Mumbai city and requires hotel service of Mumbai city frequently. Thus user's location affinity is Mumbai city. In this model, we have considered that universe is divided into the continents, continents are divided into countries, countries are divided into the states, states are divided into the district and districts are divided into cities and villages respectively. This model uses various attributes to distinguish among countries, states, cities and villages. As per our model city is specialized geographical area of the district and has several villages. Similarly, state is an aggregation of all the districts which belongs to the same state. In this model, we have not distinguished different types of states for simplicity. We have not further detailed city and village to make the model simple.Fig1 is representing the meta-model of location affinity. We are going to use this model in web service composition. In web service composition several operators are used (sequential, parallel, loop and switch). Service may be composed one after other which is called as sequential execution of the services or two services may execute simultaneously in parallel mode. One service may execute more than one time or under any defined condition. The proposed model has some pre-assumptions for service composition based on our proposed geographical criteria's. If two services are executing in the same state than their affinity is considered to be the city or district affinity. Similarly, if these services are composing from different states than their affinity is considered as state affinity. Consider that service1's location affinity is Allahabad city of Uttar Pradesh state and services2's location affinity is the Banaras city of the state Uttar Pradesh. Then, while composing their affinity is city since lying in the same state or region of our model. Similarly, we have considered that service1's location affinity is Allahabad city of Uttar Pradesh state and Service 2's location affinity is Mumbai city of the State Mumbai then compositely their affinity would be state since lying in different boundaries of our model.

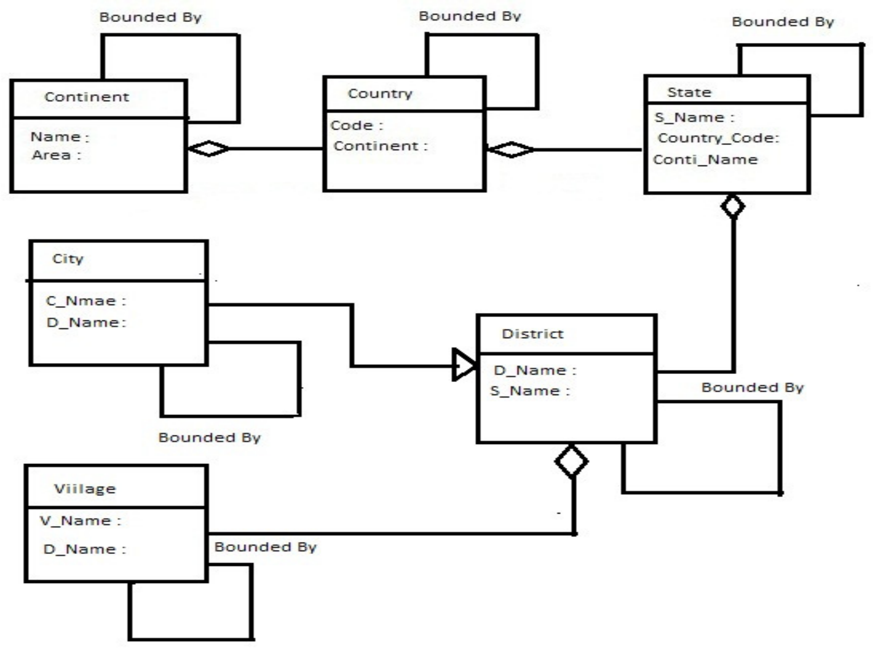

Figure 1. Modeling Location Affinity

4. Finite State Machine For Representing Web Services

Every web service consists of several input and output operations. A service changes its state due to the invocation of various available operations. These invocations may affect the parameters of quality of service attributes. For example: if the invocation is successful then reliability of web service will be higher. The other attributes also may change due to the state change like cost. For example: In ticket Reservation system if we invoke bookTicket operation, in this scenario the cost of the web service may change. In the case of single web service execution, the space related QoS attributes like affinity will not change. Several research efforts have been made to incorporate time-based QoS attributes to model web service using finite state machine [4], [24]. We are arguing that the space related QoS attribute will play a dominant role in the case of web service composition. We have modeled a web service as finite state machine that includes set of input operations, initial state, final state, transition function and quality of service parameters. Our model is based on time as well as space related QoS attributes, so it is a vector which has two different types of values one is based on geographical location and other is based on time. These component services have been modeled using finite state machine concept. A component web service is a finite state machine having touple (I, O, F, S_0, S_F, δ, Q).Where

- I is representing the set of input.
- O is representing the set operations
- δ is the transition function that is represented as :
  $\delta: I \times O \times S \times Q \rightarrow S$
- S_F is the final state of the finite state machine.
- F is the subset of S and represents the set of final states.
- Q is the quality of service factor which is categorized on the basis of the time and space dimensions i.e. Qs ε Q and Q_T ε Q as well as
  $Q = Q_T \times Q_S$
  $Q_T = Q_E \times Q_R \times Q_A$

Where Q_E is the execution time, Q_R is for reliability and Q_A for availability.

Affinity is the quality of service attribute based on location of the web service. It is computed by the user's usability of web service based on user's location preferences.

5. Modeling Composition Of Web Services Based On Location Affinity

In [4] a finite state machine based model is proposed to evaluate QoS parameters of service composition. They have also elaborated how machine learning algorithms can be used for evaluating QoS of service composition. [25] have proposed a model for synthesizing finite state machine. We are synthesizing finite machine based on time as well as space based QoS parameters.

A service could individually serve the user or it could be the composition of two or more services. Thus, overall system output is dependent on the successful execution of the each component web service existing in composition plan.

To give rules for composition of web services we are considering two state machines M1 and M2 i.e. M_1= (I_1, O_1, F_1, S_{01}, S_{F1}, δ_1, Q_1) and M_2 = (I_2, O2, F_2, S_{02}, S_{F2}, δ_2, Q_2). Resultant machine can be expressed as, M = (I, O, F, S_{01}, S_{F1}, δ, Q) .Where

$I = I_1 \cup I_2$
$O = O_1 \cup O_2$
$F = F_1 \times F_2$
$\delta: I_1 \times I_2 \times O_1 \times O_2 \times Q \times S_1 \times S_2 \rightarrow F_1 \times F_2$
$Q \rightarrow Q_1$ composition Q_2

In composition model services may execute in sequential manner, parallel manner, in the form of loop and choice [24], [29].

Thus quality of service attributes composition depends upon the composition operators. We are using the composition rules for all the composition operators as given in [24], [29]. We are summarizing these rules as given below with respect to the two services S_1 and $S_{2.}$

We have assumed that (r_1, A_1, E_1) is the reliability, availability and execution time of the service S_1 and (r_2, A2, E_2) is the reliability, availability and execution time of the service S_2 respectively.

Consider that services S_1 and S_2 are executing in sequential manner than according to the rules given in [24],[29] resultant reliability is the product of reliabilities of both the services, resultant availability is the product of availabilities of both the services and resultant execution time is the summation of the execution times of both the services respectively. Our affinity rule suggests that if two services have same level of affinity than the resultant affinity will be one level higher than the previous level affinity.

$QT=QT_1 \times QT_2= (r_1, A_1, E_1), (r_2, A_2, E_2) =(r, A, E)$
$r= r_1 \times r_2$
$A=A_1 \times A_2$
$E=E_1+E_2$

Consider that services S_1 and S_2 are executing in parallel manner than according to the rules [24,29] resultant reliability is the minimum of the reliabilities of both the services, resultant availability is the minimum of the availabilities of both the services and resultant execution time is the maximum of the execution times of both the services respectively. If the parallel execution of web services is taking place during composition than the resultant affinity level is considered to be at same affinity level.

$QT=QT_1 \parallel QT_2 = (r_1,A_1,E_1)\parallel((r_2,A_2,E_2)=(r,A,E)$
$r = r_1\parallel r_2=\text{minimum } (r_1, r_2)$
$A=A_1\parallel A_2=\text{minimum } (A_1, A_2)$
$E=\text{maximum } (E_1, E_2)$

Consider that services S_1 and S_2 are executing in loop than according to the rules given in [24,29] resultant reliability is the minimum of the reliabilities of both the services, resultant availability is the minimum of availabilities of both the services and resultant execution time is the multiple of the number of times loop is executing a service respectively. In loop based execution of web services the resultant affinity level is considered to be as before the execution of the service.

$QT=QT_1 \text{ loop} QT_2 = (r_1,A_1,E_1)\text{loop}((r_2,A_2,E_2)=(r,A,E)$
$r= r_1= r_2$
$A= A_1=A_2$
$E=n \times E$

Where n is the number of times loop is executing.

Consider that services S_1 and S_2 are executing based on choice than according to the rules [24],[29] resultant reliability is the minimum of the reliabilities of both the services, resultant availability is the minimum of availabilities of both the services and resultant execution time is the maximum of the execution time of both the services respectively. In choice operator the affinity is the affinity of the web service that is selected for the execution among all available choices.

$Q_T=QT_1$ choice $QT_2 = (r_1,A_1,E_1)$or($(r_2,A_2,E_2)=(r,A,E)$
$r = \min(r_1, r_2)$
$A=\min(A_1, A_2)$
$E=\max(E_1, E_2)$

Example 1.

Consider the Figure 2, which represents the composite execution of the purchase order service. This service uses four services for completing the execution plan. These services are defined as:

Buyer:

This service is agent service that initiates the composition plan and invokes seller service after giving the request of particular product to purchase.

Seller:

This service receives the item request from Buyer service and process the request by initiating the execution of InventoryCheck and CreditCheck services. After receiving the response returned by both the services Seller acknowledge the Buyer about the status of his request of product purchase.

InventoryCheck:

This service is responsible for checking the quantity of the product ordered by the Buyer. This service ensures that the quantity of item requested by the Buyer is available in the stock, if not it sends a negative acknowledgement to the Seller service to maintain the consistency.

CreditCheck:

This service checks the Buyer’s balance and acknowledge the seller, whether Buyer is having sufficient amount to purchase the item or not.

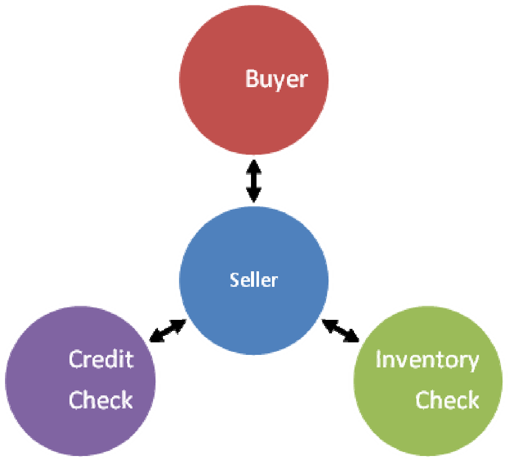

Figure 2. Composite Purchase Order Service

5.1. Aaa Finite State Machine for Buyer Service

Finite state machine takes the name of the product as an input and request product order. In this description of the finite state machine pro_name is representing name of the product ordered by Buyer. Here reqOrder () operation is used to send the product order request to the Seller. Few variables are also used to represent the communication process among services. In this finite state machine representation poreqstate variable is used to store the status of the request whether it is sent from the Buyer's end to the Seller's end or not and r_1,a_1,e_1,la_1 variables are used to store reliability, availability, execution time and location affinity of the Buyer service.

- I= {pro_name}
- O= {reqOrder ()}
- δ = {pro_name} × {reqOrder()} × S_{11}{poreqstate, r_1, a_1,e_1, la_1}$\rightarrow$$S_{12}${poreqstate ="Sent", r_1=.7, a_1=.8 ,e_1=8, la_1= "Lucknow"}
- Q_T= {.7, .8, 8}
- Q_S= {"Lucknow"}

5.2. Bbb Finite State Machine for Seller Service

Finite state machine of Seller service takes the name of the product and quantity of the product as an input. In this description of the finite state machine pro_name and quantity are representing name of the product and quantity of the product ordered by Buyer. processOrder () operation is used for further processing in the composite system. Variables poreqstate, ccrequeststate, ivreqstate,poack are used to store the status of the product order request state, credit check request state, inventory check request state In this scenario.r_2,A_2,E_2,la_2 variables are used to store reliability, availability, execution time and location affinity of the Seller service.

- I= {pro_name, quantity}
- O= {processOrder ()}

- δ={pro_name,quantity}×{processOrder()}×S_{21}{poeqstate,ccrequeststate,ivreqstate,poack, r_2,A_2,E_2,la_2}$\rightarrow$$S_{22}${poreqstate="received",ccreqstate="sent",ivreqstate="sent",poack="sent", r_2=.60,A_2=.55,E_2=5,la_2= "Allahabad" }
- Q_T= {.60, .55, 5}
- Q_S= {"Allahabad"}

5.3. Ccc Finite State Machine for InventoryCheck Service

Finite state machine takes the quantity of the product as an input. checkInvAvailability () operation is used to check whether the required quantity of the product is available in inventory stock or not. In this context ivreqstate, invavailablestate, invack, r_3, A_3, E_3, la_3 notations are used to describe inventory request state, inventory availability state, inventory acknowledgement state, reliability, availability, execution time, location affinity of InventoryCheck service.

- I= {quantity}
- O={checkInvAvailability()}
- δ = {quantity}× {checkInvAvailability()} × S_{31}{ ivreqstate ,invavailablestate,invack,r_3,A_3,E_3,la_3} $\rightarrow$$S_{32}${ivreqstate="received",inavailablestate="true",invack="sent", r_3=.7,.A_3=.6, E_3=7, la_3= "Mumbai" }
- Q_T= {.7, .6, 7}
- Q_S= {"Mumbai"}

5.4. Ddd Finite State Machine for CreditCheck Service

Finite state machine takes the account detail of the buyer as an input. In this finite state machine checkAvailability () operation is used to check the required balance for purchasing the product. In this scenario ccreqstate, ccavailablestate, ccack, r_4, A_4, E_4, la_4 notations are used to describe credit check request state, credit availability state, credit acknowledgement state, reliability, availability, execution time, location affinity of Credit Check service.

- I= {account_detail}
- O={creditAvailability ()}
- δ = {account_detail}× {checkAvailability() }× S_{41}{ ccreqstate,ccavailablestate, ccAck,r_4,A_4,E_4,la_4}$\rightarrow$ S_{42}{ccreqstate="recieved",ccavailablestate="true",ccack="sent",r_4= .9,A_4= .8 ,E_4=11,la_4=" Banaras" }
- Q_T= {.9,.8,11}
- Q_S= {"Banaras"}

5.5. Eee Service Composition

These Services have been composed to complete the system execution. In this example Buyer service is executing first in sequential manner with Seller service. Seller service than invoke the InventoryCheck and CreditCheck services in parallel to give desired output.

Buyer.Seller.InventoryCheck||CreditCheck

The resultant Quality of service and finite state machine of composite service purchase order can be represented as given below. Where $I_{B.S}$, $Q_{TB}.Q_{TS,}$ $Q_{SB}.Q_{SS}$ are representing input , time based QoS, space based QoS of Buyer andSeller together.

- $I_{B.S}= \{I_B \text{ U } I_S\}$ = {pro_name, quantity}
- $O_B .O_S= \{O_B\text{U}O_S\}$ = {reqOrder (), processOrder ()}
- $\delta_B.\delta_S=I_B\times I_S\times O_B\times O_S\times$ S11$\times S_{21}\rightarrow S_{12}\times S_{22}$= {pro_name} × {quantity} × {reqOrder()} × { processOrder()} $\times S_{11}$\{poreqstate , r_1 , A_1 , E_1 , la_1\} × S_{21}\{poreqstate , ccrequeststate , ivreqstate , poack , r_2 , A_2 , E_2 , la_2\} → S_{12}\{poreqstate="Sent", r_1=.7, A_1=.8, E_1=8, la_1="Lucknow"\} × S_{22}\{poreqstate="received", ccreqstate="sent", ivreqstate="sent", poack="sent", r_2=.6, A_2=.55, E_2=5, la_2=" Allahabad"\}
- $Q_{TB}.Q_{TS}$= {.42, .44, 13}
- $Q_{SB}.Q_{SS}$ = {"STATE AFFINITY"}

Since Inventory and Credit Check services are executing in parallel manner
$W_I||W_C$

Finite Sate machine for composition have been represented as:

- $I_{CI.CC}= \{I_{CI} \text{ U } I_{CC}\}$ = {quantity, account_detail}
- $O_{CI.CC}= \{O_{CI} \text{ U } O_{CC}\}$ = {checkInvAvailability() , checkAvailability ()}
- $\delta_{CI} .\delta_{CC}=I_{CI}\times I_{CC}\times O_{CI}\times O_{CC}\times$ S31$\times S_{41}\rightarrow S_{32}\times S_{42}$={Quantity} × { Account_detail } × { checkInvAvailability()} ×{checkAvailability ()} × S_{31} { ivreqstate , invavailablestate, invack , r_3, A_3, E_3, la_3} × S_{41}\{ ccreqstate , ccavailablestate, ccack , r_4 , A_4 , E_4 , la_4\}→S_{32}\{ivreqstate="received", inavailablestate="true", invack="sent", r_3=.7, A_3= .6, E_3=7 , la3= "Mumbai"\} × S_{42}\{ccreqstate="received", ccavailablestate="true", ccack="sent" , r_4= .9 , A_4= .8 , E_4=11 , la_4="Banaras" \}
- $Q_{TS}.Q_{TC}$={ .7, .6, 11 }
- $Q_{SI}.Q_{SC}$= {"STATE AFFINITY"}

In this finite state model $I_{CI.CC}$, $O_{CI.CC}$, $\delta_{CI} .\delta_{CC}$, $Q_{TI}.Q_{TC}$, $Q_{SI}.Q_{SC}$ are representing the composite input, output, transition function ,time based QoS and space based QoS of the composite Inventory and Credit check services. Complete composite finite state machine for purchase order service is given below:

- I= {pro_name, quantity, account_detail}
- O= {reqOrder (), processOrder (), checkInvAvailability(),checkAvailability ()}
- δ = {pro_name, quantity, account_detail}×{ reqOrder (), processOrder (), checkInvAvailability(), checkAvailability()} $\times S_{1f}$\{poreqstate, r1, A1, E1, la1, ccrequeststate, ivreqstate, poack, r_2, A_2, E_2, la_2, ivreqstate , invavailablestate, ccreqstate, ccavailablestate, ccack, r_4, A_4, E_4, la_4, invack, r_3, A_3, E_3, la_3\} →S_{2f}\{ poreqstate="Sent", r_1=.7,A_1=.8,E_1=8, la_1="Lucknow", poreqstate="received", ccreqstate="sent", ivreqstate="sent", poack="sent", r_2=.6,A_2=.55,E_2=5,la_2= "Allahabad", ivreqstate ="received", inavailablestate="true", invack="sent",r_3=.7, A_3=.6, E_3=7, la_3= "Mumbai", ccreqstate="received" , ccavailablestate="true" , ccack="sent",r_4= .9,A_4= .8, E_4=11, la_4="Banaras"\}
- $Q_{TB}.Q_{TS}$= {.29, .26, 24}

- $Q_{SB}.Q_{SS}$ = {"COUNTRY AFFINITY"}

Thus
of the service having similar functionality and similar affinity than we send the replica copy for the location affinity of the complete composite service execution is considered to be "Country" and in case of the failure service providing the same functionality from the country affinity is suggested to replace with failed service.

6. REPLICATION POLICY

In the view of advancement in distributed computing, need of replication is also growing rapidly Service replication is also the most popular remedy to avoid failures during service execution. In the scenario where a system contains only three and four services full replication gives 90% fault tolerance. In case of the system having thousands of services, replication is not a feasible choice for us due to storage and processing overhead.

So in distributed computing full replication is a very costly affair. It generally added measurable amount of cost in total composition cost. We are providing a methodology according to which only one service from the same functionality domain as well as location affinity is replicated to reduce the storage overhead of the system. Replication of a service is done on the basis of the threshold value. This threshold value is considered as the popularity factor of the service from the same location affinity. This threshold value is calculated on the basis of the service usage pattern of the user.

Definition 1. Popularity index is the index according to which replica of a service is being created. This popularity

is dependent upon the average usage amount of the service by the users in the System. Notation for popularity factor is P and U for average service usage amount.

$$P \propto U \qquad (1)$$

$$P = RU \qquad (2)$$

Where R is the constant and depends upon the ratio of the number of users of service and total users of the system. [28] have also modeled replication manager concept to assign jobs to the web services based on round robin algorithm for proper utilization of resources. In this research we are proposing replication manger concept to calculate the popularity index and creating replica of most popular service.

Consider a system that contains the five users of the seller services from the same location affinity that is Mumbai (Maharashtra).There are five seller services from the same location affinity that are providing similar set of functions to the user. These services are $Seller_1$, $Seller_2$, $Seller_3$, $Seller_4$, and $Seller_5$ respectively. In the system there are 500 users. Replicating all the services will increase the performance as well as storage overhead .Solution of this problem is the calculation of popularity factor. Service having highest population factor is being replicated to avoid the fault in the system.

Table 1. Service Usage Frequency

Service /User	**$Seller_1$ (MUM)**	**Seller $_2$ (MUM)**	**$Seller_3$ (MUM)**	**$Seller_4$ (MUM)**	**$Seller_5$ (MUM)**
$User_1$ (MUM)	100	100	300	100	100
$User_2$ (MUM)	200	50	100	100	100
$User_3$ (MUM)	300	30	300	100	50
$User_4$ (MUM)	50	150	100	100	50
$User_5$ (MUM)	100	50	50	100	100

Popularity factor is calculated for all the service given in table 1.

$P_{SELLER1}$ = (5/500)×(750/5)=1.5
$P_{SELLER2}$ = (5/500) ×(380/5)=.76
$P_{SELLER3}$= (5/500) × (850/5) =1.7
$P_{SELLER4}$= (5/500) × (500/5) =1
$P_{SELLER5}$= (5/500) × (400/5) =.80

From the calculations given above Seller3 is identified as the highest popularity service. This service is replicated in the distributed environment. Periodic update in the service usage pattern is performed by replication manager to find out the highest popular service in the service domain and particular location affinity.

6.1. Aaa PROPOSED ALGORITHM

Taking in to consideration the concept given in the replication policy we are assuming that we have already a replica of the service providing similar services from the same affinity as a failure handling mechanism. Our suggested algorithm is taking three types of the recovery decisions .Firstly it checks the replica copy of the failed service if we found the replica execution. If it found replica copy as unavailable as it takes the replacement approach as a remedy of failure. Lastly if both the foresaid operations are unsuccessful it just roll-back the execution of the service.In this scenario r_1,a_1,E_1, is representing the reliability, availability ,execution time of the failed service and r_2,a_2,E_2 reliability, availability ,execution time of the service to be replaced.

Algorithm 1. Service Recovery Based on time as well as space QoS

Input: Failed Service
Output: Selected Recovery Mechanism

```
1.    Check for the availability of the replica of the service.
2.    If (Replica Available) then
3.    Provide replica of the most popular service providing similer functionality from the same
      location affinity to the user.
4.    Else
5.    If (r1<r2 && a1<a2 && E1 >E2&& user's
      location affinity==location Affinity of Service)
      Then
6.     replace the service
7.    Else
8.    Rollback the failed service
```

7. CONCLUSION

In this paper we have modeled geographical locations for computing location affinity of the services. We have also integrated time based QoS attributes and space based QoS attributes to model failure. A service failure recovery policy is also introduced that takes recovery decisions dynamically based on our suggested QoS attributes. Replication is the basic demand of distributed systems but it is a costlier effort to implement. Thus we have initially included a popularity value to decide the service that should be replicated. We also suggested the algorithm that takes recovery decisions based on time as well as space based QoS parameters.

REFERENCES

[1] Sunil R Dhore, M U Kharat,"QoS Based Web Service Composition using Ant Colony Optimization: Mobile Agent Approach", International Journal of Advance Research in Computer and Communication Engineering, vol. 1, issue 7, pp. 519-527, September 2012.

[2] Zhou Xiangbing, Ma Hongjiang, Miao Fang "An Optimal Approach to the QoS-Based WSMO Web Service Composition Using Genetic Algorithm", Workshops Lecture volume 7759, pp 127-139, Springer, 2013.

[3] Quan Z. Sheng a, xiaoqiang Qiao b,, Athanasios V. Vasilakos c, Claudia Szabo a,Scott Bourne a, xiaofei xu d ,"Web services composition: A decade's overview", Information Sciences 280 ,pp. 218–238,Elsevier,2014.

[4] Olga kondratyeva,Natalia Kushik ,Ana Cavalli,Nina Yevtushenko," Evaluating Web Service Quality using Finite State Models", Proc. 13th International Conference on Quality Software, pp.95 – 102, IEEE, 2013.

[5] Kevin Wiesner, Roman Vacul´ın, Martin Kollingbaum, and Katia Sycara, "Recovery Mechanisms for Semantic Web Services", Distributed Applications and Interoperable Systems ,Lecture Notes in Computer Science, volume 5053, pp 100-105,Springer,2008.

[6] Thirumaran.Ma,Dhavachelvan.Pb,Shanmugapriya.Rc, Kiran Kumar Reddyd," A Novel Approach for Web Service Run Time Exception Handling", In the proc. of 2nd International Conference on Communication, Computing & Security, volume 6, pp.145–152, Elsevier,2012.

[7] Jocelyn Simmonds, Shoham Ben-David, Marsha Chechik," Guided Recovery for Web Service Applications", Proc. eighteenth ACM SIGSOFT international symposium on foundations of software engineering, pp. 247-256, 2010.

[8] Suchi Gupta, Praveen Bhanodia,"A Fault Tolerant Mechanism for composition of Web Services using Subset Replacement", International Journal of Advance Research in Computer and Communication Engineering, vol. 2, issue 8, pp. 3080-3085, August 2013.

[9] Jocelyn Simmonds, Shoham Ben-David, Marsha Chechik, "Monitoring and Recovery of Web Service Applications", volume 95,issue 3,pp. 223-267,computing ,2012.

[10] Rafael Angarita, Yudith Cardinale, Marta Rukoz,"Reliable Composite Web Services Execution: Towards a Dynamic Recovery Decision", Electronic Notes in Theoretical Computer Science, vol.302, pp.5-28, Elsevier, 2014.

[11] Hadi Saboohi, Sameem Abdul Kareem," Requirements of a Recovery Solution for Failure of Composite Web Services", International Journal of Web & Semantic Technology, vol.3, issue 4, pp. 3-19, October 2012.

[12] Zho Wu,Naixue xiong,Wenlin Han,Yan N. Huang,Chun Y. Hu,Qiong Gu, Bo Hang, "A Fault-Tolerant Method for Enhancing Reliability of Service Composition Application in WSNs Based on BPEL", International Journal of Distributed Sensor Networks ,volume 2013,arcticle id 493678,II page.

[13] Hossein Rahmani, Hassanabolhassani, "Composite Web Service Failure Recovery Considering User Non-Functional Preferences ", Proc. of 4th International Conference on Next Generation Web Services Practices, pp. 39 – 45, IEEE, 2008.

[14] H. Elfawal Mansour and T. Dillon, Fellow, IEEE," Dependability and Rollback Recovery for Composite Web Services", IEEE TRANSACTIONS ON SERVICE COMPUTING, vol. 4, pp. 328 – 339, October-December 2011.

[15] Keting Yin , Bo Zhou , Shuai Zhang , Bin Xu "QoS-Aware Services Replacement of Web Service Composition", Proc. of Internal Conference on Information Technology and Computer Science, pp.271 – 274, IEEE,2009.

[16] Guisheng Fan, Huiqun Yu, Liqiong Chen, Chunhua Gu1," An Approach to Handling Failure Recovery in Service Composition and Its Analysis", Proc. of Fifth IEEE International Conference on Theoretical Aspects of Software Engineering, pp. 153 – 160,2011.

[17] Abdelkarim Erradi, Piyush Maheshwari2, Vladimir Tosic,"Recovery Policies for Enhancing Web Services Reliability", Proc. IEEE International Conference on Web Services, pp. 189 – 196, 2006.

[18] Cao Jiuxin, Zhou Tao, Zhu Gongrui, Liu Bo, Luo Junzhou, "Execution Recovery in Transactional Composite Service", Proc. 20th International Conference on Web Services, pp.276-283, IEEE,2013.

[19] Joyce El Haddad, Maude Manouvrie, Marta Rukoz, "TQoS-Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition", IEEE Transactions on Services Computing, vol. 3, issue1, pp.73-85, 2010.

[20] Mohamed Sellami, Samir Tata, ZakariaMaamar, and Bruno Defude,"Recommender System for Web Services Discovery in a Distributed Registry Environment". Proc. Fourth International Conference on Internet and Web Applications and Services, pp.418 - 423, IEEE, 2009.

[21] Marwa F. Mohamed, Hany F, ElYamany, Mohamed K. Hussien, Nashwa, M. Yhiea2 and Hamed M. Nassar,"An Adaptive Replication Framework For Improving The QoS Of Web Services", vol. 2, issue 1,SpringerPlus ,2013.

[22] Mario Bravetti, Stephen Gilmore, Claudio Guidi , and Mirco Tribastone, " Replicating Web Services for Scalability", Proc. of Third Symposium on Trustworthy Global Computing Lecture Notes in Computer Science, Springer, Volume 4912, 2008, pp. 204-221.

[23] Ivona Brandic, Sabri Pllana, Siegfried Benkner, "High-level composition of QoS-aware Grid workflows: An approach that considers location affinity", Proc. Workflows In Support Of Large-Scale Science, pp. 1 - 10, IEEE, 2006.

[24] R.S. Pandey, B.D. Chaudhary, "An estimation of min-max of QoS attributes of a choreography", In proc. of International Conference on Advances in Engineering, Science and Management , pp. 872 - 878 ,IEEE ,2012 .

[25] Jun Sun, Jin Song Dong,"Design Synthesis from Interaction and State-Based Specifications ", Transactions on Software Engineering, vol. 32, issue 6, pp. 349 - 364 IEEE, 2006.

[26] An Liu, Qing Li, Senior Member, IEEE, Liusheng Huang, and Mingjun Xiao, "FACTS: A Framework for Fault-Tolerant Composition of Transactional Web Services", IEEE Transactions On Services Computing, vol. 3, issue 1, pp. 46 - 59, 2010.

[27] Johannes Behl, Rüdiger Kapitza, Matthias Schunter , "Providing Fault-tolerant Execution of Web-service–based Workflows within Clouds ", Proc. 2nd International Workshop on Cloud Computing Platforms. Bern, Switzerland, article no. 7, ACM, 2012.

[28] Pat. P. W. Chan, Michael R. Lyu," Reliable Web Services: Methodology, Experiment and Modeling", Proc. IEEE International Conference on Web Services, pp. 679 - 686 2007.

[29] Huiyuan Zheng, WeiliangZhao, Jian Yang , Bouguettaya, A.," QoS Analysis for Web Service Compositions with Complex Structures" , IEEE Transactions on Services Computing, vol. 6,issue 3, pp.373 – 386,2007.

AUTHORS

Dr. R. S. Pandey PhD. (Computer Science and Engineering) from Motilal National Institute of Technology (MNNIT). M.Tech. (Computer Science and Engineering) from Motilal National Institute of Technology (MNNIT). M.Sc (Computer Science) from J.K Institute of Applied Physics (Allaahabad University) . Working as Assitant Professor at Birla Institute of Technology since 2000.

Richa Pathak M.Tech.(Computer Science and Engineering) from Birla Institute of Technology.MCA from U.P. technical University.Worked as Lecturer at Shri Ramswaroop Memorial College of Engineering and Technology,Lucknow (U.P.Technical University) from 2010 to 2013.