International Journal of Grid and Distributed Computing
Vol. 9, No. 10 (2016), pp.385-396
http://dx.doi.org/10.14257/ijgdc.2016.9.10.34

Research on Probabilistic Optimization to Dynamic Composition for Service Replacement

Honghao Gao[1,2], Minjie Bian[2], Yucong Duan[3] and Yonghua Zhu[2]

[1]Computing Center, Shanghai University, 200444 Shanghai, P.R. China
[2]School of Computer Engineering and Science, Shanghai University, 200444 Shanghai, P.R. China
[3]College of Information Science and Technology, Hainan University, 570100 Haikou, P.R. China
gaohonghao@shu.edu.cn

Abstract

A growing number of enterprises have been moving their works to encapsulate system functions, business logics and processing modules into Web service because of its flexibility and low-cost. However, service-oriented software calls for constantly adjusting its architecture in order to respond to varying user requirements and instable runtime environments. One of the most challenging issues is how to effectively implement a reconfiguration to ensure the business-critical application is trustworthy. In this paper, it proposes a method to dynamic composition for service replacement, which focuses on the probabilistic optimization to service planning of candidate compositions when the service failure is occurred. First, the input and output data specification is defined to describe interface behaviors, and then the probabilistic solution graph is introduced to formalize replacement strategies. Second, corresponding algorithms are discussed for optimization selection purpose, which includes reliability calculation process and model modification process. The former computes the probability value of each service planning generated from probabilistic solution graph. The latter modifies probabilistic solution graph model to recommend Top-k solutions, pruning the service planning which does not satisfy the specified probability value. Third, the architecture of prototype is presented to demonstrate the feasibility of the proposed method. Our method provides a reference to guarantee the reliability of service process in E-commerce.

***Keywords**: Dynamic Composition, Service Replacement, Probabilistic Solution Graph, Quantitative Computing, Optimized Service Planning*

1. Introduction

The solution of service oriented architecture (SOA) has become an important method to achieve resources sharing and applications integration through services composition [1-3]. In general, service process is defined in term of workflows, a flexible mechanism to implement business logics [4], by which the advantages of high interoperability, cross-platform-ability and loose coupling contribute to dynamically assign services to each task of workflow in run-time environment. However, due to the heterogeneous, open and collaborative natures of Internet, any service failure may immediately result in operation faults and application interrupts [5]. It will seriously influence on the correctness and reliability of business process, even causing economic losses.

There are two categories of service failures, which are mainly caused by internal factor and external factor. For the former, service lacks required resources, which limits activities to further execution. For the latter, the instable Internet makes the random failure phenomenon frequently happened in service process. Thus, the quality of service

ISSN: 2005-4262 IJGDC

(QoS) is treated as an important issues in software system [6-8]. It is no guarantee that a service can always be executed correctly, which should adapt and react to changings from time to time. The feasible way to handle the failed service is service replacement [9-20], including one-to-one mode and one-to-more mode. The one-to-one mode replacement substitutes the failed service with a new service selected from service repository, using backups predefined in configuration files. However, it is an error-prone and time-consuming since the manual definition depends on the scale and complexity of services. If there is no suitable candidate service, the services composition has been proposed to support service replacement by integrating a set of services. Therefore, the one-to-more mode performs services composition in dynamic process, which selects a number of services to properly work together performing tasks of the failed service. This replacement is worked in the form of service planning [10-12], where each service planning can accomplish functional requirements of the failed service. The dynamic composition gives all possible services planning according to business logics of the failed service.

In order to guarantee service planning is an efficient manner, it should consider following problems. 1) How to build a formal model for service planning? Modeling service planning into a solution graph is regarded as replacement strategies. However, there is no standard work on modeling service planning at present. 2) How to compute the reliability of each service planning in a qualitative way? As a non-functional characteristic of Web services, the formal model should be extended with probabilistic behaviors to the reliability evaluation. User often prefers to select optimal service planning to replace the failure service. For instance, the service planning with the high reliability will be prior to recommended to user [13-14].

To this point, a probabilistic optimization approach is proposed to service replacement for dynamic composition. The main contributions of this paper includes: 1) A probabilistic solution graph model is used to construct all possible service planning with probability description. The services matching method is based on interface analysis. 2) A corresponding algorithm is discussed for Top-k service planning recommendation purpose, focusing on user requirements with the service reliability and service number. It considers composing an optimal service planning.

The rest of this paper is organized as follows. Section II introduces our motivation. Section III gives formal models for probabilistic solution graph. Section IV discusses corresponding algorithms to recommendation purpose. Section V presents an architecture introduction to show each module function. Section VI reviews related works. Section VII concludes this research and discusses future directions.

2. Motivation

Suppose that there is an equipment management process which covers all phases of device management life cycle, including an application preprocessing AP service, a purchase device PD service, an in-stockroom operation ISO service, and a scrap handling SH service. If one service is unavailable, the equipment management process will be crash which calls for service replacement by dynamically selecting related services. The new created large-granularity service is regarded as a candidate to substitute the failed service. For example, selected services, including an ordering OB service, an online payment OP service, and an express delivery ED service, will be integrated into a composite service to simulate behaviors of the purchase device PD service.

However, services composition is a highly complex task. First, it demands to select appropriate services for dynamic planning. Web service publishes its function via interface for user or third-party application to be invoked. How to select and organize them together into a composite service requires understanding their functions and operation differences because some services provide same functionalities with different interface parameters, such as input data and output data. For example, we may select another planning for handling purchase device PD service, such as composing a booking payment BP service and an express delivery ED service. Second, it is important to check whether the input and output interface of a composite service satisfy the failed service or not, which should make sure that the dynamic composition is worked under the correct direction. For example, online payment service NOP is selected from several candidates if and only if it can cooperate well with order booking service and express delivery service. However, suppose that service NOP has two outputs while express delivery service calls for three inputs, the online payment service NOP should be removed from dynamic composition and reselect a new online payment service with three outputs. Third, each service displays different reliabilities since Internet is open and changing. The probabilistic behavior has become an important evaluation point to different Web services. The probability value not only dependents on service itself but also correlates to Internet environment. The dynamic composition may have different service planning schemes under different probability values. The Top-k solutions with optimal probability should be paid more attentions. For example, user preference requires getting a composite service with the minimum service number. Obviously, service planning *sp* including BP service and ED service is better than service planning *sp*' including OB service, OP service and ED service. However, if the former has a lower reliability, the latter is considered as optimal solution.

The above sources of complexity for service replacement are attributed to dynamic composition issues. To address the challenge, the first problem is how to construct model for service planning when a service failure is occurred. Considering the function simulation, it requests composite service to provide same function to failed service. Thus, it should study the function guided service planning according to interface parameters. The second issue is how to select the suitable service planning when considering both user preference and service reliability. Service planning composed by different number services will display different probability values, which may take different costs and response time. Although the composition result with the minimum service number is useful, there is a need to develop a method to effectively recommend Top-k solutions focusing on the probability value of each service planning.

3. Formal Modeling and Probabilistic Solution Graph

In this section, we first define key concepts for service specifications, and then propose a probabilistic solution graph to modeling service planning when services composition is requested to respond to the failed service.

Definition 1 (Web Service). A Web service is defined as a tuple $ws=(I,O,F,D,G,P)$, where,

1) $I=\{i_1,i_2, i_3,.., i_n\}$ is a set of input parameters.

2) $O=\{o_1,o_2, o_3,.., o_n\}$ is a set of output parameters.

3) F is the functional description of each Web service.

4) D is the domain concept, which represents a function classification.

5) $G=(id,name,location,provider)$ is the background information, such as identifier, name, location, provider.

6) $P \in [0,1]$ is a probability value which denotes the reliability of Web service.

In Definition 1, each service is identified by symbol G describing basic information, including service location, service provider and service name. Symbol *id* is used as a unique identifier, and F describes service functions. Symbol P is a probability value to denote the reliability of Web service, which is computed by failure statistics. Symbol D is a precondition to service replacement for dealing with the failed service, which requests candidate services should be belonged to the same domain. Without the help of domain, it is difficult to find appropriate services when searching functional description with keyword. To this purpose, we introduce Ontology tree to describe service domain.

Definition 2(Ontology Tree). The hierarchical relation is employed to specify the knowledge classification, which is implemented by Ontology tree. It is defined as a tuple $OT=(C, R)$, that,

1) C is a set of concepts represented by nodes, where each node is mapped with services.

2) R is a set of direct inheritance relationships represented by edges.

Suppose that two concepts $ws_1.D_1=c_1 \in C$ and $ws_2.D_2=c_2 \in C$. In Ontology tree, there are two equivalences. If they are belonged to same concept, then the equivalence relationship is $c_1 = c_2$ called as the absolute equivalence. If c_1 inherits from c_2 that $R(c_1, c_2)$, then the equivalence relationship is $c_1 \approx c_2$ called as the simulation equivalence.

Definition 3 (The Compatible Services). Two services ws_i are compatible, $i \in \{1,2\}$, if and only if they have an intersection between the output of former services ws_1 and the input of latter service ws_2. It is defined as $ws_1 \propto ws_2$, that.

$$ws_1.O \cap ws_2.I \neq \varnothing \qquad (1)$$

We use symbol ~ to denote sequence composition of two compatible services that $ws_1 \propto ws_2$. For example, ws_1~ws_2 is a composite service composed of ws_1 and ws_2 in sequence. The service planning ws_1~ws_2 can be used to replace the ws when they are located at same domain, that $ws_1 \propto ws_2$ and ($ws_1.D = ws.D$ and $ws_2.D=ws.D$, or $ws_1.D \approx ws.D$ and $ws_2.D \approx ws.D$). The relation is defined as follow.

$$ws_1 \sim ws_2 \models ws$$

$$\equiv\ ws_1 \propto ws_2 \wedge (ws_1.D = ws.D \wedge ws_2.D=ws.D) \vee (ws_1.D \approx ws.D \wedge ws_2.D \approx ws.D\ ) \qquad (2)$$

The dynamic composition is to build a solution graph by organizing a set of services which consists of multiple strategies to implement the input and output behaviors. The service planning should display same input and output behaviors comparing with the failed service.

Definition 4 (Interface Data). The interface data is a group of interface parameters including input I and output O used by Web service. It is defined as $P=\{p_1,p,p_3,..,p_n\}$ where $p_i \in P$ is a name-value paired attribute variable.

Two interface behavior types {↓, ↑} are proposed to specify service interaction patterns. Symbol ↓ is used to label the input behavior that $I=P{\downarrow}=\{a_1,a_2,a_3,..,a_n\} \subseteq P$, and symbol ↑ is used to label the output behavior that $O=P{\uparrow}=\{b_1,b_2,b_3,..,b_n\} \subseteq P$. Symbol $|P|$ is used to compute the element number of interface data set, which is called as the length of interface data.

In order to select and compose services in a well-behaved manner, there are two concepts about data matching, mainly data equivalence and data consistency. The data equivalence is used to compare the selected service with failed service focusing on interface data equivalence at global level. While the data consistency is used to check whether two selected services is compatible or not at local level during services composition for dynamic planning.

Definition 5 (Data Equivalence). Given two interface data A and B, they are data equivalence $A \cong_e B$ if and if only two interface data satisfy following rules,

1) The interface data sets A and B belong to the same interface behavior pattern.

2) $|A|=|B|$, where the length of each data set is equal.

3) $\forall i \bullet a_i=b_i \wedge a_i \in A \wedge b_i \in B$.

Based on Definition 5, the composite service should be data equivalence to the failed service, not only input parameters but also output parameters. Suppose composite service cs and failed service fs are data equivalence. After replacing the failed service using service planning, the data equivalence should be $cs.I \cong_e fs.I$ and $cs.O \cong_e fs.O$.

$$cs \cong_e fs \equiv cs.I \cong_e fs.I \wedge cs.O \cong_e fs.O \quad (3)$$

The data consistency is used as a guidance to service selection. Thus, individual services for service planning can be dynamically selected to satisfy data equivalence according to the interface decomposition.

Definition 6 (Data Consistency). Given two interface data $A\uparrow$ and $B\downarrow$, the data consistency of two interfaces $A\uparrow \cong_c B\downarrow$ is defined as follows,

1) The interface data sets A and B belong to the different interface behaviors.

2) $|A|=|B|$, where the length of each data set is equal.

3) $\forall i \bullet a_i=b_i \wedge a_i \in A \wedge b_i \in B$.

Let $ws_1 \sim ws_2 \sim ws_3 \sim \ldots \sim ws_{n-1} \sim ws_n$ be a service planning. The data consistency requires each two connected services pair (ws_i, ws_{i+1}) should satisfy $ws_i.O \cap ws_{i+1}.I \neq \varnothing$ and $ws_i.O \cong_d ws_{i+1}.I$, that,

$$\forall i \in N, \exists (ws_i, ws_{i+1}) \bullet ws_i.O \cap ws_{i+1}.I \neq \varnothing \wedge ws_i.O \cong_d ws_{i+1}.I \quad (4)$$

In addition, it is also data equivalence when the current service planning is used to replace the failed service fs that $ws_1.I \cong_e fs.I$ and $ws_n.O \cong_e fs.O$.

$$ws_1 \sim ws_2 \sim ws_3 \sim \ldots \sim ws_{n-1} \sim ws_n \cong_c fs$$

$$\equiv ws_1.I \cong_e fs.I \wedge ws_n.O \cong_e fs.O \quad (5)$$

In order to get a set of services to simulate the interface behaviors, the service planning is considered to find a path compositing services to be executed in sequence, which calls for both data consistency and data equivalence during dynamic composition. Given a failed service ws, there may be different services composition for service planning. Thus, we propose probabilistic solution graph model to formalize service planning for all possible services compositions.

Definition 7 (The Probabilistic Solution Graph Model Service Planning). Suppose a failed service fs needs to be replaced. The finite state machine (FSM) is extended with data to model service planning. It is a tuple $PSG=(S, s_0, s_f, \delta, L, A, P)$, where,

1) S is a finite set of services.

2) $s_0 \in S$ is the initial state, and $s_f \in S$ is the finial state.

3) $\delta \subseteq S \times S$ is a set of transition relations, where two compatible services $ws_1 \sim ws_2$ are connected as a transition relation.

4) A is a set of interface data specifying input and output parameters.

5) $L(S) \rightarrow 2^A$ is the labeling function, denoting the data used by each state $s \in S$.

6) $P(s) \in [0,1]$ is a probability value.

In probabilistic solution graph, it is a directed graph. The initial state and finial state follow the data equivalence that $fs.I \cong_e s_0.I$ and $fs.O \cong_e s_f.I$. The labeling function denotes each state with interface data specifications, which requires the data consistency between neighboring two services.

The path set $Tran= \{<s_0,s_1,..,s_f> | \exists s_i, s'_{i+1} \in S,\ i<n \bullet \bigwedge_{i=0}^{n-1} (s_i, s'_{i+1}) \in \delta\}$ consists of all possible service planning for simulating behaviors of the failed service. Each path is called as service planning because it can implement the input and output behavior. The model is not empty if and only if $PSG \neq \varnothing$, that,

$$Tran= \{<s_0,s_1,..,s_f> | \exists s_i, s'_{i+1} \in S,\ i<n \bullet \bigwedge_{i=0}^{n-1} (s_i, s'_{i+1}) \in \delta\} \neq \varnothing \quad (6)$$

There may be multiple paths with different probability values that can be adopted to fulfill functions of the failed service. The probability value of reliability for each service planning $sp=<s_0,s_1,..,s_f> \in Tran$ is the aggregated probability value of selected services, which is computed as follow.

$$P(sp) = P(< s_0, s_1,.., s_n >) = \prod_{i=0}^{n} P(s_i) \qquad (7)$$

Although each service planning actually provides the same functionality, they have different reliability. The maximal probability value for a service planning *sp* is computed by

$$P^{max}(Tran)=Max\{P(sp_i)\}\ where\ \forall i\bullet sp_i\in Tran \qquad (8)$$

However, user would like to choose service planning with the minimum service number since the cost of selected services may be much cheaper. However, it is hard to make sure the service planning with the minimum service number has the highest probability value. Each service of service planning acts as a whole which will impact the dynamic composition result. Thus, we should make a balance between the probability value and service number. Furthermore, some service planning in probabilistic solution graph should be pruned if the probability value does not satisfy the specified value.

4. Algorithms for Modeling Service Planning and Probabilistic Optimization

In this section, we briefly introduce our main algorithms. The first algorithm to modeling service planning is discussed. The second algorithm to Top-k solutions recommendation for probabilistic optimization is introduced.

4.1. Modeling Service Planning Algorithm

Definition 8 (Service Repository). The service repository stores a set of services, which is defined as follow,

$$SR = \{ws_i\}_{i=0}^{m} \qquad (9)$$

Definition 9 (Service Selection). The service selection is to select corresponding services to dynamic composition for the failed service. The service selection set is denoted as

$$SS = \{ws_j\}_{j=0}^{n} \quad where \quad n \le m \qquad (10)$$

In order to make sure that the service selection set *SS* satisfies the domain of the failed service, the checking process is worked as follow,

$$\forall ws_i \in SS\Box ws_i.D = fs.D \vee ws_i.D \approx fs.D \qquad (11)$$

The service planning is reduced to give a sequence composition according to the input and output parameters of the failed service.

$$SP = \{< s_0, s_1, s_2, ..., s_f > | (\exists s_i, s_{i+1} \in SS\Box s_i.O \cong_d s_{i+1}.I) \wedge (s_0.I \cong_e fs.I \wedge s_f.O \cong_e fs.O)\} \ (12)$$

In Table 1, it is the service planning modeling algorithm *SPAM*(). The inputs of *MSPA* are failed service *fs* and service repository *RS*. The outputs are formal models in the form of probabilistic solution graphs.

Table 1. Service Planning Modeling Algorithm

***Algorithm**: Service Planning Modeling Algorithm SPMA()*
***Input**: The Failed Service fs and Service Repository RS*
***Output**: Probabilistic Solution Graph Model PSGs*

Function SPMA (Failed Service fs, Service Repository RS)

```
Extract domain of the failed service fs that D←fs.D
For each element service ws in RS
  If ws.D = D or ws.D ≈ D Then
     SS ← ws
     If $ws_i.I \cong_e fs.I$ Then
        Stack $Ts_0 \leftarrow ws_i$
     If $ws_i.O \cong_e fs.O$ Then
        Stack $Ts_f \leftarrow ws_i$
  End If
End For
For each element service $s_0$ in $Ts_0$
  For each element service $s_f$ in $Ts_f$
    For each element service ws in SS
       $SP=\{<s_0,s_1,s_2,...,s_f>|(\exists s_i,s_{i+1}\in SS\square s_i.O\cong_d s_{i+1}.I)\wedge(s_0.I\cong_e fs.I\wedge s_f.O\cong_e fs.O)\}$
    End For
  End For
End For
Construct PSGs that each PSG=(S, $s_0$, $s_f$, $\delta$, L, A,P) according to SP
Return PSG
End Function
```

In service planning modeling algorithm, the first step is to extract domain *D* of the failed service to recognize the service classification. The service selection set *SS* is used as candidate services to dynamic composition, during which the initial state and finial state are stored into stack Ts_0 and Ts_f respectively. The second step is to traverse stacks Ts_0 and Ts_f for building a pair (s_0,s_f) to figure out the service planning *SP* according to formula (12). After obtaining *SP*, the algorithm returns multiple probabilistic solution graphs as outputs.

4.2. Probabilistic Optimization Algorithm

In the process of probabilistic optimization, the first step aims to prune the formal model of probabilistic solution graph, where the service planning without satisfying specified requirements will be cut. The second step is to give the Top-k solutions from the modified probabilistic solution graph model.

How to select service planning to handle the failed service is changed to the Top-k solutions recommendation problem. However, before selecting planning strategies, it should remove services that the corresponding planning does not satisfy the limitations of user preferences [15].

Definition 10 (The Modified Probabilistic Solution Graph Model). The modified probabilistic solution graph model is generated from probabilistic solution graph model, in which each service planning meets user requirements. The user preferences is defined as (η, ρ) where η stands for the service number and ρ represents the reliability. The modified probabilistic solution graph model is denoted as $PSG_r \ \Xi \ PSG$ where

1) $S_r = S / T$ where $T \subseteq Tran$ is a subset of paths *Tran*

2) $s^r{}_0=s_0$, $s^r{}_f= s_f$

3) $\delta_r \subseteq S_r \times S_r$

4) $A_r \subseteq A$

5) $L_r(s)=L(s)\rightarrow 2^{Ar}$ and $P_r(s)=P(s) \in [0,1]$ where $s \in S_r$

The subset paths for pruning probabilistic solution graph model is computed by comparing with (η, ρ) that,

$$T=\{t|t\in Tran\bullet|t|>\eta\vee P(t)<\rho\} \tag{13}$$

Thus, when $T\neq\varnothing$, *PSG* model should be modified. The *S* / *T* means the partition results that if path sequence $\exists t\in T$ has a component appeared in safety path sequence $t'\in Tran\text{-}T$ then the component service is called as mixed service, which should be retained. The modification process is to remove services which are not mixed.

Definition 11 (Top-k Solutions of Service Planning). Let *SP* be a set of service planning, and *fs* be a failed service, where each element $t\in SP$ can be regarded as a composite service. The Top-k solutions of service planning satisfy the following conditions.

1) $t.I \cong_c fs.I \wedge t.O \cong_c fs.O$

2) |SP|=k

3) $\forall t\in SP, \forall t'\in(Tran_{all}\text{-}SP)\bullet\ t\succ t'$

Symbol $\succ$ means "better than" which demands at least one factor is optimal. For example, two service planning t$\succ$t' that $t\in SP$ and $t'\in(Tran_r\text{-}SP)$ should be P(t)>P(t') or |t|<|t'|. Symbol $Tran_{all}$ is the set of all path sequences that are modified from probabilistic solution graph models *PSGs*.

In Table 2, it is the probabilistic optimization algorithm *POA()*. The inputs of *MSPA* are probabilistic solution graph models *PSGs* and user requirements (η, ρ). The outputs are Top-k solutions of service planning *SP*.

Table 2. Probabilistic Optimization Algorithm

***Algorithm**: Probabilistic Optimization Algorithm POA()*
***Input**: Probabilistic Solution Graph Models PSGs and User Requirements (η, ρ)*
***Output**: Top-k Solutions Service Planning SP*

```
Function POA (Probabilistic Solution Graph Model PSGs and User Requirements ($\eta$, $\rho$))
For each PSG $\in$ PSGs
    Construct $Tran= \{<s_0,s_1,..,s_f>|\exists s_i,s'_{i+1}\in S, i<n \bullet \bigwedge_{i=0}^{n-1} (s_i,s'_{i+1})\in\delta\}$
    $T=\{t|t\in Tran\bullet|t|>\eta\vee P(t)<\rho\}$
    Use S/T to cut the mixed service belonged to T, generating $PSG_r$ that $PSG_r$ $\Xi$ PSG
    $Tran_{all}\leftarrow Tran(PSG_r)$
End For
If $P^{max}(Tran_{all})<\rho$ Then
   Return NoPlanning
End If
For i=1 to k
  For each $t\in Tran_{all}$
    Log=true
   For each $t'\in(Tran_{all}\text{-}t)$
      If $t \nsucc t'$ that $\neg(P(t)>P(t')\vee|t|<|t'|)$Then
        Log=false
       Break
     End If
   End For
  If Log=true Then
     $SP\leftarrow t$
     $Tran_{all}\leftarrow Tran_{all} / t$
  End If
End For
End For
End Function
```

In probabilistic optimization algorithm, the first step is to construct *Tran* for getting a set of service planning. When user requirements are inputted, the service planning with a unsatisfied probability value $P(t)<\rho$ or service number $|t|>\eta$ will be removed from the probabilistic solution graph. The modification work is to cut the unmixed service in order to break the service planning. After that, the remaining service planning is collected into a new container $Tran_{all}$. In second step, it uses a loop with k times to select Top-k service planning. The result guarantee Top-k solutions SP to satisfy $\forall t \in SP, \forall t' \in (Tran_{all}-SP) \bullet t \succ t'$ where each service planning is better than the remaining solutions in $Tran_{all}-SP$.

5. Architecture and Discussion

The framework to dynamic composition for service replacement is called as DCSR, which includes composition modeling and computing processes, mainly four parts that the service selection, service planning modeling, service planning checking, and service planning recommendation. As Figure 1, shown, the aim of DCSR is to provide an efficient way to replace the failed service when service software is running.

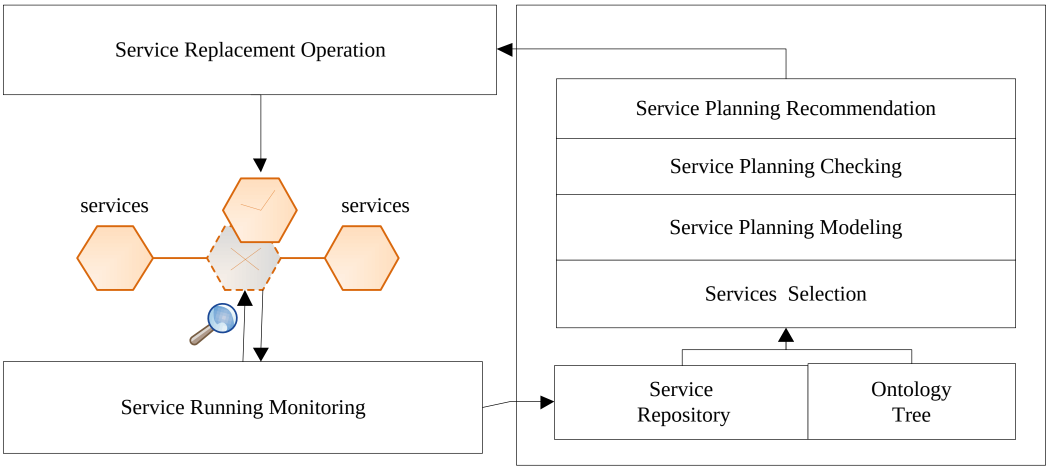
Service Replacement Operation
services
services
Service Running Monitoring
Service Planning Recommendation
Service Planning Checking
Service Planning Modeling
Services Selection
Service Repository
Ontology Tree

Figure 1. The Architecture of Service Replacement

In service repository, Web service is specified by XML-based languages, which is extended from WSDL for adding extra information descriptions. The Ontology tree generated by Protege in the form of OWL is used as service domain, which requires each service should be mapped to a node concept of Ontology tree. The service running monitoring is responsible for monitoring service based software. It is implemented by history log focusing on error code analysis and statistics. If the service failure is detected, then the service replacement will be requested.

The service selection is designed to select candidate services from service repository. The domain of the failed service is used as selection principle. Services belonged to the same domain or superclass domain will be selected from service repository. The selection result is called as candidate services set.

The service planning modeling is in charge of formalizing probabilistic solution graph model. It executes dynamic composition according to interface decomposition strategy. The service planning result satisfies both data consistency and data equivalence. The former monitors the composition process of two compatible services, while the latter checks the composite service against the failed service.

The service planning checking is responsible for pruning probabilistic solution graph model when user requirements are inputted. Each service planning is checked by comparing the service number and probability value. The result is used as reference to probabilistic solution graph model modification.

The service planning recommendation is responsible for choosing Top-k solutions. Each service planning is better than remaining solutions in the modified probabilistic solution graph model. Finally, the service replacement operation will execute a service planning to substitute the failed service.

6. Related Works

Web service is a way to share various types of services and resources. The dynamic composition provides a value-added process through the seamless integration of a set of services. It is considered as service replacement method to handle service failure in order to revert back to normal logic business.

To the best of our knowledge, few literatures have studied the service replacement problem. We give a review on the major techniques and researches that are most closely related to our works. Kim [16] focused on the availability of Web service and proposed a technique to improve services composition reliability using WS-BPEL. Xin [17] presented a service replacement method based on BPEL-described workflow. The relationship between services in BPEL determines the scope of service replacement. Wijesiriwardana [18] presented a dependency-based component service recommendation approach, considering the dependencies among component services, such as input/output, pre/post conditions, and correlation among services. Saboohi [19] proposed a replacement method for semantic web services using subgraph. Margaris [20] aimed to enhance adaptation quality and improve the QoS offered to end users. They discussed the identification of potential services that could be used to realize a functionality used in WS-BPEL scenarios. The transactional semantics of invocations were introduced. The above methods concern on one-to-one mode service replacement, which were reduced to single service selection when backup services are predefined. However, the Internet environment is uncertainty, which calls for dynamic composition to replace the failed service.

The dynamic composition approach to service planning induces situation calculus, state-space search, problem deduction, and automatic theorem proving. Kil [21] studied the web services composition problem considering semantic aspects, *i.e.*, exploiting the semantic relationship between parameters of Web services. They tried to find the optimal composition that contains the shortest path which satisfies the requirement. Phan [22] proposed a formal approach to translate OWL-S Web service descriptions into primitive and complex actions of ConGolog, a high level logic programming language with sensing actions for Web services composition. And an extended version of the middle-ground ConGolog interpreter was discussed. Lama [23] presented an optimal and complete algorithm to find all valid compositions from the point of view of the semantic input-output message structure matching. The solution was improved using a backward heuristic search based on A* algorithm which could find all the possible solutions with different number of services and runpath. Paik [24] proposed a framework combining logical composition (using HTN) and physical composition (using CSP) for automatic services composition. Kun [11] proposed a model of combining a Markov decision process model and HTN planning to address Web services composition. In the model, HTN planning was enhanced to decompose a task in multiple ways and hence was able to find more than one plan, taking both functional and non-functional properties into account. Rao [25] introduced a method for automatic composition of semantic Web services using linear logic theorem proving, which used DAML-S for external presentation of Web services, while, services were presented by extra logical axioms and proofs in linear logic.

However, these works lack considering the random phenomenon. The reliability computing of dynamic composition for service replacement has been growing in importance [26-27]. In this paper, we propose an approach to probabilistic optimization for dynamic composition to prune useless service planning from probabilistic solution graph model and recommend Top-k solutions.

7. Conclusion

Web service is a promising technology to the distributed software integration, where the service failure is an important research topic in services computing. Considering there is no backup service to implement replacement operation, it is necessary to study dynamic composition technology to selects proper services and integrate them together to simulate behaviors of the failed service.

To handle this problem, this paper proposes an approach about probabilistic optimization to dynamic composition for service replacement. The first process aims to generate service planning and give a formal model to formalize our method. It is called as probabilistic solution graph model, which is dynamically constructed according to the interface decomposition strategy. The second process modifies probabilistic solution graph model in order to satisfy user requirements, recommending Top-k solutions of service planning to target replacement for the failed service.

In the future, we will carry out experiments to estimate the effectiveness of our approach and consider designing an integrated platform for business applications. Furthermore, this paper only studies the reliability with probability value. It will take the response time and cost of Web service into consideration because these factors request the multiple targets optimization and performance analysis.

Acknowledgements

This paper is supported by National Natural Science Foundation of China (NFSC) under Grant No.61572306, 61502294, Natural Science Foundation of Shanghai under Grant No.15ZR1415200, Natural Science Foundation of Hainan under Grant No.20156234, CERNET IPv6 Innovation Project under Grant No.NGII20150609, 2015 Film Summit Science Foundation of Shanghai University, and Foundation of Science and Technology Commission of Shanghai Municipality under Grant No.14590500500.

References

[1] H.Groefsema and N. V. Beest, “Design-Time Compliance of Service Compositions in Dynamic Service Environments. The 2015 IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA)”, **(2015)**, pp.108-115.

[2] N. Xi, C. Sun, J. Ma and Y. Shen, “Secure service composition with information flow control in service clouds”, Future Generation Computer Systems, vol. 49, **(2015)**, pp. 142-148.

[3] Q. Liang, J. Chung and S. Miller, “Modeling semantics in composite Web service requests by utility elicitation. Knowledge and Information Systems, vol. 13, no. 3, **(2007)**, pp. 367-394.

[4] X. Fu, T. Bultan and J. Su, “Analysis of interacting BPEL Web Services”, The 13th international conference on World Wide Web (WWW), **(2004)**, pp. 621-630.

[5] Y. Yin and S. G. Deng, “Analysing and determining substitutability of different granularity Web services”, International Journal of Computer Mathematics, vol. 90, no. 11, **(2013)**, pp. 2201-2220.

[6] S. Y. Hwang, H. Wang, J. Tang and J. Srivastava, “A probabilistic approach to modeling and estimating the QoS of Web Service-based workflows”, Information Science, vol. 177, no. 23, **(2007)**, pp. 5484-5503.

[7] J. A. Parejo, S.Segura, P. Fernandez and A. R. Cortés, “QoS-aware web services composition using GRASP with Path Relinking”, Expert Systems with Applications, vol. 41, no. 9, **(2014)**, pp. 4211-4223.

[8] C. Mao, J. Chen, D.Towey, J. Chen and X. Xie, “Search-based QoS ranking prediction for web services in cloud environments”, Future Generation Computer Systems, vol. 50, **(2015)**, pp. 111-126.

[9] E. Elabd, E. Coquery and M. S.Hacid, “Compatibility and Replaceability Analysis of Timed Web Services Protocols”, The Second International Conference on Computer and Electrical Engineering, vol. 2, **(2009)**, pp.15-19.

[10] J. Puttonen, A. Lobov, M. A. C. Soto and J. L. M. Lastra, “Planning-based semantic web service composition in factory automation”, Advanced Engineering Informatics, vol. 29, no. 4, **(2015)**, pp. 1041-1054.

[11] K. Chen, J. Xu, S. R. Marganiec, “Markov-HTN planning approach to enhance flexibility of automatic web services composition”, The 2013 IEEE 20th International Conference on Web Services, **(2009)**, pp. 9-16.

[12] M. Kuzu and N. K. Cicekli, “Dynamic planning approach to automated web service composition.”, Applied Intelligence, **2012**, Vol.36, No.1, 1-28.

[13] R. Vaithiyanathan and T. A.Govindharajan, “User preference-based automatic orchestration of web

services using a multi-agent", Computers & Electrical Engineering, , vol. 45, **(2015)**, pp. 68-76

[14] W. Tan, Y. Fan and M. C. Zhou, "A Petri net-based method for compatibility analysis and composition of web services in business process execution language", IEEE Transactions on Automation Science and Engineering, vol. 6, no. 2, **(2009)**, pp. 94-106.

[15] B. Upadhyaya, Y. Zou, I. Keivanloo and J. Ng, "Quality of Experience: What End-Users Say about Web Services?", The 2014 IEEE International Conference on Web Services(ICWS), **(2014)**, pp. 57-65.

[16] J. P. Kim and J. E. Hong, "Dynamic Service Replacement to Improve Composite Service Reliability", The 2011 Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI), **(2011)**, pp. 182-188.

[17] H. Xin and Z. Z. De, "The research on web service replacement based on workflow", The 2014 5th IEEE International Conference on Software Engineering and Service Science (ICSESS), **(2014)**, pp. 251-254.

[18] W. Chaman, G. Giacommo and G. Emanuel, "Dependency Based Approach for Software Analysis Web Services Replacement", The 2012 19th Asia-Pacific Software Engineering Conference (APSEC), vol. 1, **(2012)**, pp. 294-299

[19] H. Saboohi, A. Amini and H. Abolhassani, "Failure recovery of composite semantic web services using subgraph replacement", International Conference on Computer and Communication Engineering, **(2008)**, pp. 489 -493.

[20] D. Margaris, P. Georgiadis and C. Vassilakis, "On Replacement Service Selection in WS-BPEL Scenario Adaptation", 2015 IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA), **(2015)**, pp. 10 -17

[21] H. Kil, W. Nam and H. Kil, "Semantic web service composition via model checking techniques", International Journal of Web & Grid Services, vol. 9, no. 4, **(2013)**, pp. 339-350.

[22] M. Phan and F. Hattori, "Automatic web service composition using congolog", IEEE International Conference on Distributed Computing Systems, **(2006)**, pp. 17

[23] M. Lama, P. R. Mier, M. Mucientes and J. C. Vidal, "An optimal and complete algorithm for automatic web service composition", International Journal of Web & Grid Services, vol. 9, no. 2, **(2012)**, pp. 1-20

[24] I. Paik and D. Maruyama, "Automatic web services composition using combining HTN and CSP", CIT **(2007)**, pp. 206-211.

[25] J. Rao and P. Küngas, "Logic-based web services composition: From service description to process model", IEEE International Conference on Web Services, **(2004)**, pp. 446-453.

[26] Y. Wang and E. P. Lim, "The evaluation of situational transaction trust in e-service environments", In IEEE International Conference on e-Business Engineering, **(2008)**, pp. 265-272.

[27] Y. Wang and K. Lin, "Reputation-oriented trustworthy computing in e-commerce environments. Internet Computing", IEEE Internet Computing, vol. 12, no. 4, **(2008)**, pp. 55-59.

Authors

Honghao Gao, received the Ph.D degree in computer application technology from the School of Computer Engineering and Science of Shanghai University, Shanghai, P. R. China, in 2012. His research interests include Web service and model checking.

Minjie Bian, is a PhD candidate in School of Computer Engineering and Science Shanghai University. His research interests include computer vision. He works at the Information Technology Office of Shanghai University during the study of PhD.

Yucong Duan, received the Ph.D degree in Software Engineering from Institute of Software, Chinese Academy of Sciences, P. R. China in 2006. He is currently a full professor and vice director of Computer Science department in Hainan University, P. R. China. His research interests include software engineering, service computing, cloud computing, and big data. He is a member of IEEE, ACM and CCF (China Computer Federation).

Yonghua Zhu, received the Ph.D degree from the School of Communication & Information Engineering of Shanghai University, Shanghai, China, in 2006. His research interests include formal method, high performance computing, communication and information engineering.