International Journal of u- and e- Service, Science and Technology
Vol.10, No.3 (2017), pp.57-66
http://dx.doi.org/10.14257/ijunesst.2017.10.3.06

A QoS-Aware Decentralized Service Composition Method in Mobile Networks

Nianhua Yang and Qi Li

School of Statistics and Information,
Shanghai University of International Business and Economics,
Shanghai 201620, China
yangnianhua@suibe.edu.cn, cn16270721@suibe.edu.cn

Abstract

In mobile networks, service composition usually depends on a predefined composition model or a centralized communication node. These conditions are usually difficult to be satisfied in mobile networks. This paper proposes a dynamic service composition method in mobile networks. It does not depend on previous conditions. The service composition scheme for a task is generated dynamically according to the environment of mobile networks in real-time. Furthermore, paths can be selected according to the execution load, and parameters can be adjusted personally by the user according to the environment.

Keywords: *service composition, decentralize, QoS aware, mobile networks*

1. Introduction

With the developing of various networks, service oriented computing has been widely applied for the loose coupling and platform-independent characteristics. A variety of professional service oriented applications are emerging. A service usually performs a specific function. Appropriate service composition can realize a complex business process. The mechanism of combining two or more services to form a complex service for a specific task is known as service composition [1].

A lot of services among the network usually can provide the same function with different QoS (Quality of Service, QoS). Even providing the same function, different services often require different input parameters and provide different output types. So appropriate service selection and composition models have been focused in service oriented computing area. Generally, service composition mechanisms can be divided into two categories: plan based composition and semantic based composition [2]. Plan based composition is also regarded as static composition and semantic based composition is usually regarded as dynamic composition.

In a static service composition mechanism, services are composed through three phases: workflow design, service binding and execution. However, a device, may be a service provider, could join and leave the mobile networks at any time. So the abstract workflow can't be predefined. Because services or service providers in the mobile network can't be predicted in the workflow design phase. Furthermore, some service composition requirements can't be known in advance. Plan based mechanism is not applied in mobile networks.

In a dynamic service composition, services are composed on demanded in real-time, not configured or deployed in advance [3]. Dynamic service composition can also be divided into centralized and decentralized mechanisms. In the centralized scenarios, a device is selected as a server. A new service should be registered on the server. A customer should send the request to the server for service discovery or service composition. The server will dynamically form a composed service according to the

ISSN: 2005-4246
Copyright ⓒ 2017 SERSC

reality scenarios. Although centralized approaches present simple solutions, they suffer from scalability, bottlenecks and single point of failure [2]. What is more serious is that a central server can't be assigned in advance in mobile networks. So a lot of decentralized dynamic service composition approaches [2-4] emerged for the mobile networks in recent years. In the decentralized scenarios, there is no service register center. A task is decomposed into a lot of services dynamically according to the real-time network scenarios. A task still can be finished through composing alternative services when a device is out of service for resource limitation or mobility.

In the dynamic service composition environment, there are multiple optional paths for a task. Each path is composed by different services. Though each path can achieve the same function to accomplish the task, the QoS of each path is different. The QoS of a path is determined by each service along the path. The QoS of each service or the whole path is determined by different elements, such as execution time, success rate, service price, *etc.* Different people have different preferences to the QoS elements. Some people expect the shortest execution time. Others expect the highest execute success rate. Still others expect the lowest cost. So the system should automatically select the optional composition path according to the customer's personal preference.

A lot of customers will require services in the network. A service may be selected by different tasks. Multi users may wait for executing the same service and others which can provide the same functionality may be vacant. The QoS of a service may be changed under different execution load.

This paper proposes a personalized QoS aware dynamic service composition method in a mobile network. A task is decomposed into services according to those available in the network. The task requestor will compare QoSs of different candidate path and select the best one to be executed. The weight for different elements of QoS is assigned by a requestor personally. Execution load is regarded as a dynamic element of QoS in dynamic network environment.

The main contributions of this paper include: (1) a personalized QoS-aware dynamic service composition method is proposed, (2) a dynamic task decomposing algorithm is proposed for dynamically producing distributed execution path, (3) load balance is considered for execution path planning.

The remainder of this paper is organized as follows. Section 2 discusses related work. QoS model is presented in Section 3. Section 4 presents QoS and load balance aware personalized dynamic service composition method. Section 5 discusses the proposed method. Section 6 offers conclusions and suggestions for future work.

2. Related Work

Service composition techniques can be classified into plan based and semantic based mechanisms. Plan based composition is usually denoted as a workflow which indicates transitions between a lot of states. Each transition is bound with a service statically. The workflow designing and service binding are usually determined previously. A lot of plan based service composition mechanisms [5-6] have been proposed in the recent years. However, services which provide proper functions are usually not known in the design stage in a mobile network. Furthermore, a device which provide services may leave the mobile network at any time. So a plan based service composition mechanism can't provide a reliable solution. On the other hand, a service with better QoS may be added into the mobile network at run time. How to dynamically bind a better service to improve adaptability of a workflow has attracted a lot of interests. Some solutions [7-8] have been proposed to improve adaptability of workflows. However, these solutions require central entities for service composition. Providing a central entity is not practical in a mobile network. Lee *et al.* [9] propose a functional flexibility service composition method to satisfy different QoS goals. But this method requires a predefined workflow which is not suitable for a mobile network.

Semantic based service composition mechanisms facilitate discovering, composing and executing services dynamically according to the real-time scenarios. They have attracted much attentions [1-4, 10, 11] for their high adaptability. Fujii *et al.* [12] propose a service composition approach based on the semantics of services and components. They also combine components based on semantic and users' contexts [13]. The mechanisms proposed by Fujii *et al.* [12, 13] rely on a central middleware or manager. Kalasapur *et al.* [1] treat the service composition as a problem of finding a shortest path between two services in a directed acyclic graph. This approach also needs a central registry. A semantic based service discovery is proposed by Paliwal *et al.* [11]. Unfortunately, this method also relies on a centralized registry. Tong *et al.* [6] propose an automatic service composition method. It requires an initial offline human intervention to generate plans. So it is inadaptable to highly dynamic environments in the mobile networks for the static nature of plans.

Al-Oqily *et al.* [3] propose a decentralized service composition solution. However, service discovery in this solution relies on the geographic location of each node in the network. But we can't assume that every node in the network equipped with geographic location sense module. Chen *et al.* [4] propose a dynamic service composition model for adaptive systems in mobile computing environments using a semantic based overlay network to discover logical neighbors instead of geographic ones. The execution plan is formed dynamically. But an abstract workflow of the task should be defined in advance. Furthermore, this approach is not QoS aware. Zhu *et al.* [14] propose a load aware dynamic service selection model. In the model, the abstract workflow should also be defined in advance.

In this paper, we propose a novel decentralized dynamic service composition mechanism. It can procedure a workflow of a task dynamically according to the environment. Personalized QoS and service execution load are regarded as measurements for service selection.

3. System Model

This section firstly proposes a QoS model for service description. Then, the service model is proposed based on functionality, inputs, outputs, QoS, *etc.*

3.1. QoS Model

Each service has some unique properties such as average execution time, mean time between failures, cost, service execution waiting queue length, *etc.*

Def. 1 QoS properties of a service is a quintuple $QoS = (SID, AET, MTBF, C, L)$, where SID represents the identification of the service. AET represents average execution time of the service. $MTBF$ represents mean time between failures. C represents the cost of each execution. L represents a list of requestors who is waiting for execution.

The execution load of a service can be calculated by Eq. (1).

$$Load_{service} = \sum_{i=1}^{len(L)} AET_i \tag{1}$$

Eq. (1) adds each AET of the service for requestors. So the length of L and each AET reflect the execution load of a service.

A node in the network may provide more than one service. The load of a node can be calculated by Eq. (2).

$$Load_{node} = \sum_{j=1}^{numberOfServices} \sum_{i=1}^{len(L)} AET_{ji} \tag{2}$$

Eq. (2) adds all loads of services which provided by a node.

3.2. Personalized QoS Calculation Method

Different people have different attentions to the elements of QoS. For example, cost may the determinant for a person, but for others, AET may be the determinant for service selection.

So the integrative QoS value of a service will influence the selection. An integrative QoS value is calculated by Eq (3).

$$q = \alpha \cdot v_{AET} + \beta \cdot v_{MTBF} + \gamma \cdot v_{Cost} + \delta \cdot v_{Load_{node}} \tag{3}$$

In the Eq. (3), the larger of the q means the better for the quality of service. α, β, γ and δ represents the weights for the properties of AET, MTF, execution cost and execution load of the node. Their values are assigned by the requestor and meet the condition that $\alpha + \beta + \gamma + \delta = 1$.

AET, MTF, cost and $Load_{node}$ has different measurement unit with each other. So the value of them should be normalized for integration of these values. v_{AET}, v_{MTF}, v_{Cost} and $v_{Load_{node}}$ are their normalized values respectively. The normalization methods are described in Eq. (4)(5)(6)(7).

$$v_{AET} = \frac{AET_{\max} - AET}{AET_{\max} - AET_{\min}} \tag{4}$$

$AET_{\min}$ can be set to be zero. A requestor sets the value of $AET_{\max}$ for the whole network. If the AET of a service is larger than $AET_{\max}$, the service will not be included in the path. Once the values of $AET_{\min}$ and $AET_{\max}$ are specified, the normalized value v_{AET} will be increased with the reduce of AET.

$$v_{MTBF} = \frac{MTBF - MTBF_{\min}}{MTBF_{\max} - MTBF_{\min}} \tag{5}$$

$MTBF$ is defined as the average duration the service can work correctly between failures. The bigger of $MTBF$, the better of the service. $MTBF_{\min}$ and $MTBF_{\max}$ can be assigned by the requestor as the personalized parameter acting on the entire network. The normalized value v_{MTBF} will increased with the increase of $MTBF$ under the specified $MTBF_{\max}$ and $MTBF_{\min}$.

$$v_{\cos t} = \frac{C_{\max} - C}{C_{\max} - C_{\min}} \tag{6}$$

$C_{\min}$ can be set to be zero. $C_{\max}$ can be assigned by the requestor as the personalized parameter acting on the entire network. The normalized value $v_{\cos t}$ will decreased with the increase of C.

$$v_{Load_{node}} = \frac{Load_{node}^{\max} - Load_{node}}{Load_{node}^{\max} - Load_{node}^{\min}} \tag{7}$$

In Eq. (7), the parameters represent the load for a node which may provide more than one services. $Load_{node}$ is calculated by Eq. (2). $Load_{node}^{\min}$ can be set to be zero. $Load_{node}^{\max}$ can be assigned by the requestor as the personalized parameter acting on the entire network. The normalized value $v_{Load_{node}}$ will decreased with the increase of $Load_{node}$.

3.3 Task and Service Model

A task is handled by a service composition system which receives the task's specification as an input and composes concrete services [4]. In this paper, a task is specified as $T = (TID, I_T, O_T, Q_T)$, where TID is the identification of the task, I_T and O_T are input and output sets of the task, Q_T is the QoS limitation which the requestor can tolerated. $Q_T = (AET_{\max}, MTBF_{\min}, C_{\max}, Load_{node}^{\max})$, where $AET_{\max}$, $MTBF_{\min}$, $C_{\max}$ and $Load_{node}^{\max}$ have been described in previous.

A service is specified as $S = (NID, SID, I_S, O_S, Q_S)$, where NID represents the node identification who provides the service, SID represents the service identification, I_S represents the required input set of the service, O_S represents the output set of the service, Q_S represents the QoS properties of the service. $Q_S = (AET, MTBF, C, Load_{service})$. Q_S is updated dynamically for the value of $Load_{service}$ is changed in real-time.

4. Decentralized Dynamic Service Composition Method

For inherent dynamic environment in mobile networks, predefined abstract workflow for a task is not adaptable. In this section, we propose an approach to decomposing the task into concrete services according to the real-time environment of the network. Within a period of time, the requestor will get several candidate execution paths. Each path is composed by some concrete services. The path with greatest integrate QoS value will be selected as the execution path. The others will be regarded as alternates. During execution, the load of each node on the path will be updated. In this section, we provide a workflow generation method and service information updating mechanism.

The requestor will broadcast a message for service request to accomplish a task. The message is specified as $m_T = (MID, I_T, O_T, L_S, Q_T, TTL)$, where MID represents the message identification, I_T represents the required input set, O_T represents the output set of the service. L_S represents the ordered list of services along the path who can produce the expected output through cooperation. TTL represents the time to live of the message.

$Q_T = (AET, MTBF, C, Load)$. If the execute path is realized by the concatenation of services, $AET = \sum_{i=1}^{n} AET_i$, $1/MTBF = \sum_{i=1}^{n}(1/MTBF_i)$, $C = \sum_{i=1}^{n} C_i$ and $Load = \sum_{i=1}^{n} Load_i$, where n is the number of services along the execution path. If the execute path is paralleled by services, $AET = Max(AET_i)$, $MTBF = Min(MTBF_i)$, $C = \sum_{i=1}^{n} C_i$, $Load = Max(Load_i)$.

In this paper, we suppose that service has an integral output set which can't be divided, and the input can be divided into one or two sets such as shown in Figure 1. Though the input set could also be divided into multi sets, the process method is similar.

Figure 1. Service Type

Based on this assumption, there are two typical service compositions as shown in Figure 2 and Figure 3.

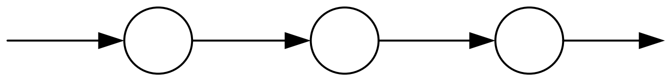
Figure 2. Concatenation of Services

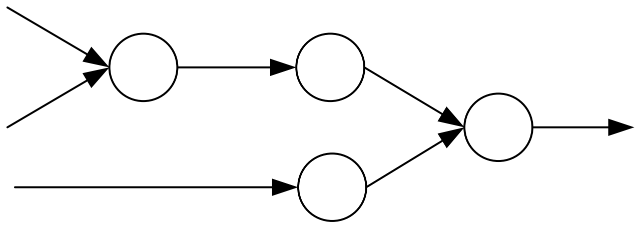
Figure 3. Parallel of Services

If a node in the mobile network needs to finish a special task, it broadcasts a request message and waits for reply messages within the message's TTL. If the requestor receives a message initiated by itself, L_s in the message will be added to possible path set L.

Once the TTL reduces to zero, or the number of the received possible paths whose input set equals to that of the initial message is large than the requestor's specified number k, the requestor will stop to receive the reply message and add possible paths whose input set equals to that of the initial message into the confirmed path set P_c.

Once the TTL reduces to zero, and there exists some possible paths whose input set is equal to that of the initial message, the requestor will add possible paths whose input set equals to that of the initial message into the confirmed path set P_c.

Once the TTL reduces to zero, and there is no possible path whose input set is equal to that of the initial message, the requestor will try to combine some possible paths in the L into a confirmed path whose union set of input ones is equal to that of the initial message. For example, Figure 4 and Figure 5 show that two paths in the possible path set L hold a same sub path from service B to service A. If the union of the sets I_{B1} and I_{B2} is equal to the input requirement of the service B, these two possible paths can be composed into an integrated service, such as shown in Figure 6. Furthermore, if the union of input sets I_1 and I_2 is equal to that of the request task, the combined execution path will be added into the confirmed path set P_c. Once the number of elements in P_c is large than k or the consuming time on combining is expired, combing process will stop.

If P_c is not null, it calculates the QoS value of each element in P_c. If some QoS values are large than Q_T, the element in P_c with max QoS value will be selected as the execution path.

Otherwise, if the time is out and the set P_c is still null, or the QoS value of each confirmed path is less than the required value Q_T, that means service composition for the required task is fail in current mobile environment. The requestor could rebroadcast the message after a while.

The detailed algorithm for previous process is described in Alg. 1.

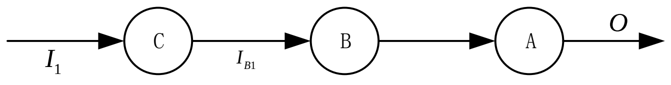

Figure 4. A Possible Path for Requestor

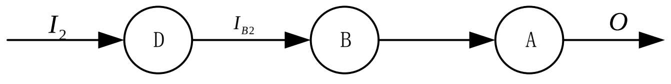

Figure 5. Another Possible Path for Requestor

Algorithm 1. Process of a service requestor in mobile networks

```
Input: $I_T, O_T, Q_T, TTL, k$
Output: service composition scheme
1.  Candidate path set $L$ is set to be null
2.  broadcasts a message $m_T = (MID, I_T, O_T, L_S, Q_T, TTL)$
3.
4.  wait for messages
5.
6.  if (the message is initiated by itself):
7.      while ($TTL > 0$ and len( $L$ )< $k$ and $L_S$ in the message is not null):
8.          add $L_s$ to $L$
9.
10. if ( len( $L$ )>0 and the input set of some paths equals to that of the initial message):
11.     add the possible path whose input set equals to that of the initial message
                into the confirm path set $p_c$
12.
13. If ( $TTL = 0$ and len( $L$ )>0 and there is no possible path whose input set equals to
                                      that of the initial message ):
14.     While (some possible paths can be integrated and len( $p_c$ )< $k$ ):
15.         Integrate some possible paths
16.         Add the integrated result in to confirmed path set $p_c$
17.
18. If ( $p_c$ is not null):
19.     Calculate $QoS$ value of each element in $p_c$
20.     Return the element in $p_c$ with max $QoS$ value as the execution path
21. Else:
22.     Return null
```

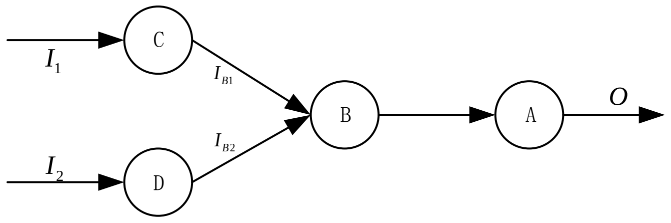

Figure 6. A Composited Path for Execution

When a mobile node receives a message $m_T = (MID, I_T, O_T, L_S, Q_T, TTL)$, it will modify the message according to the Alg. 2 and broadcast the modified message.

Algorithm 2. Process of a message receiver in mobile networks

```
Input: a message $m_T = (MID, I_T, O_T, L_S, Q_T, TTL)$
Output: a modified message
1.  If ($TTL$ >0 and $L_S$ is not null and the receiver holds a service $s$ whose output satisfies at
    least one input set of the head of $L_S$ and $s$ is not in $L_S$ ):
2.      Add $s$ as the head of the list $L_S$
3.      $I_T$ is set to be the input of the added service $s$
4.      $TTL = TTL$ -1
5.  Else if ($TTL$ >0 and $L_S$ is null and
            the receiver holds a service $s$ whose output equals $O_T$ ):
6.      Add $s$ into $L_S$
7.      $I_T$ is set to be the input of the added service $s$
8.      $TTL = TTL$ -1
9.
10. Return the modified message
```

When a service is selected to be executed to perform the task, its load is added. Similarly, when a service accomplishes the execution for a task, the load is reduced.

5. Discussion

In a mobile environment, success rate of service composition is influenced by the density of mobile users, travelling speed, personalized parameters, *etc.* In our method, users can specify the value of TTL, confirmed execution paths number k, minimum QoS requirement value, *etc.* If the service composition fails, the requestor can adjust these parameters, and rebroadcast the service composition request after a period.

6. Conclusion and Future Work

This paper proposes a personalized decentralized dynamic service composition method in mobile networks. This method avoids shortcomings of traditional methods which depend on predefined service composition models or a central node. These conditions typically can't be satisfied in mobile networks. Our method is distributed and does not depend on any predefined service composition models or a central node. Detailed algorithm for dynamic service composition in mobile networks is proposed.

In the future, we will carry out experiments to obtain reliable parameters. Furthermore, a trust model will be introduced to encourage cooperation between participants in the mobile networks. Otherwise, service composition success rate will still be low even in dense networks.

Acknowledgments

This work was partially supported by the Research Innovation Program of Shanghai Municipal Education Commission under grant No. 14YZ134, the Humanities and Social Science Research Planning Fund of the Education Ministry of China under grant No. 15YJCZH201, Shanghai 085 Project for Municipal Universities.

References

[1] S. Kalasapur, M. Kumar and B.A. Shirazi, Dynamic service composition in pervasive computing, IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 7, pp. 907-918 (2007).
[2] Y.A. Ridhawi and A. Karmouch, Decentralized plan-free semantic-based service composition in mobile networks, IEEE Transactions on Services Computing, vol. 8, no. 1, pp. 17-31 (2015).

[3] I. Al-Oqily and A. Karmouch, A decentralized self-organizing service composition for autonomic entities, ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 6, no. 1, pp. 1-18 (2011); DOI 1921641.1921648.

[4] N. Chen and S. Clarke, A dynamic service composition model for adaptive systems in mobile computing environments, Service-oriented computing, Lecture notes in computer science 8831, X. Franch, A. Ghose, G. Lewis and S. Bhiri, eds., Springer Berlin Heidelberg (2014), Vol. 8831, pp. 93-107.

[5] E. Sirin, B. Parsia and J. Hendler, Filtering and selecting semantic web services with interactive composition techniques, IEEE Intelligent Systems, vol. 19, no. 4, pp. 42-49 (2004).

[6] H. Tong, J. Cao, S. Zhang and M. Li, A distributed algorithm for web service composition based on service agent model, IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 12, pp. 2008-2021 (2011).

[7] M. Adams, A.M.t. Hofstede, D. Edmond and W.P.v.d. Aalst, Worklets: A service-oriented implementation of dynamic flexibility in workflows, On the move to meaningful internet systems 2006: Coopis, doa, gada, and odbase, Lecture notes in computer science 4275, R. Meersman and Z. Tari, eds., Springer Berlin Heidelberg (2006), Vol. 4275, pp. 291-308.

[8] L. Ardissono, R. Furnari, A. Goy, G. Petrone and M. Segnan, Context-aware workflow management, Web engineering, Lecture notes in computer science 4607, L. Baresi, P. Fraternali and G.-J. Houben, eds., Springer Berlin Heidelberg (2007), Vol. 4607, pp. 47-52.

[9] C.-H. Lee, S.-Y. Hwang and I.L. Yen, Service composition with functional flexibility using nondeterministic service interface, IEEE 10th International Conference on e-Business Engineering (ICEBE 2013), IEEE Computer Society, (2013) 11-13 Sept. 2013, Coventry, United Kingdom, pp. 435-440.

[10] R. Groenmo and M.C. Jaeger, Model-driven semantic web service composition, 12th Asia-Pacific Software Engineering Conference (APSEC '05), IEEE Computer Society, (2005) 15-17 Dec. 2005, Taipei, Taiwan, pp. 79-86.

[11] A.V. Paliwal, B. Shafiq, J. Vaidya, X. Hui and N. Adam, Semantics-based automated service discovery, IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 260-275 (2012).

[12] K. Fujii and T. Suda, Semantics-based dynamic service composition, IEEE Journal on Selected Areas in Communications, vol. 23, no. 12, pp. 2361-2372 (2005).

[13] K. Fujii and T. Suda, Semantics-based context-aware dynamic service composition, ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 4, no. 2, pp. 1-31 (2009).

[14] Y. Zhu, W. Li and J. Luo, Multi-user oriented load-aware dynamic service selection model, Ruan Jian Xue Bao / Journal of Software (in Chinese), vol. 25, no. 6, pp. 1196-1211 (2014)

Authors

Nianhua Yang, received his BSc in Management Information System from Beijing Information Technology Institute, China, in 2001, MSc and PhD in Computer Science and engineering from East China University of Science and Technology in 2004 and 2011 respectively. He is now an associate professor in the Shanghai University of International Business and Economics. His current research interests include software engineering, service oriented computing, privacy preserving *et al.*

Qi Li, received her BSc in Information Management and Information System from Shanghai University of International Business and Economics, China, in 2016. She is now a graduate student in the Shanghai University of International Business and Economics. Her current research interests include data mining and privacy preserving.

Copyright ⓒ 2017 SERSC