International Journal of Grid Distribution Computing
Vol.8, No.1 (2015), pp.101-108
http://dx.doi.org/10.14257/ijgdc.2015.8.1.12

Socially Aware Discovery Approach Supporting Service Composition

Yong Sun[1 2]

[1]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China.*
[2]*ChuZhou University, Chuzhou, 239000, China*
syong@nuaa.edu.cn

Abstract

Service composition provides a promising solution to integrate service components and business process, enabling cooperation with new partners. It increasingly relies on multiple service providers to fulfill a given complex business goal. Despite significant progress, most approaches of service composition are still impractical and almost unusable, because the current methods of service composition lack consideration of interaction among service providers. With the increasement of providers and services, it becomes difficult to find the required services in the distributed computing environment. This work addresses such issue and suggests metrics to discover the relationships of service providers by accounting of recent service interactions. In this way, a social network of service providers is constructed, and modeled as a social graph. Service clients can navigate the network and understand the structure and the metrics of social graph by employing social network analysis. Most importantly, through the constructing social networks, the efficiently service composition approach is therefore presented to discover a group of collaboration service providers for satisfying the business requirements. In addition, a case study is conducted to demonstrate the socially aware composition approach in details.

Keywords: *Service composition; Social network; Steiner Tree; Business process analysis; Social Collaboration.*

1. Introduction

Service oriented computing has been approved as the main way of software service delivery. Service composition provides a promising solution to complete the more complex business tasks [1]. The composition approaches are usually employed to integrate service components and composite business process, enabling cooperation between new partners [2-3]. It cooperatively creates values by integrating the specific services.

Service composition technologies dramatically promote the development of collaboration work among different service providers. Some tasks in business process can be outsourced to external service providers for the large scale distributed business and scientific systems [4]. A modern information application includes both internal processes and external processes serving collaboration among enterprises [5]. The partners often work together to complete a complex business process.

Despite significant progress, the current methods of service composition are still impractical and almost unusable. Service providers published their web services in UDDI based on service descriptions. The current service descriptions, such as WSDL, only consider services as isolated functional elements despite their collaborations with others before. History of successful workflow invocation isn't saved for service composition. Thus, they know only about themselves, but not

ISSN: 2005-4262 IJGDC

about the partners that they would like to collaborate in a composing way. In this situation, only few of services published on the web have already been successfully discovered and invoked.

Social collaboration is nearly ignored in the composing processes. In fact, organizations can set up and foster networks of their colleagues, customers, and providers for the business development. Capturing service interactions could be beneficial for software engineers who can capitalize on the known successful interactions. Linked social service providers can be used to overcome the drawback of the current service composition approach [6]. It is helpful to find highly potential partners based on social network analysis.

How to create the social networks of service providers becomes the basic problem of social aware service composition. Fortunately, more and more workflow applications have recorded execution trails in event logs [7]. As interactions are taken, logging and monitoring tools observe services interactions in the workflow systems [8]. We can analyze the recorded execution trails, and know how services are invoked and combined, and how their providers collaborated before. The relationships among service providers are mined by accounting of recent interactions of services. Based on the relationship information, a social network of service providers is modeled as a directed graph, where the node reflects the network member, and each edge denotes the relationships between two service providers.

It is hard to determine the optimal composite solution in the complex network for satisfying business goals. However, little work has been done on such issue, even though this is required in socially aware service composition. Therefore, in this paper, we address this issue and propose an effective social aware solution for service composition. Our proposed approach provides a mechanism, which adopts process analysis and social network analysis to overcome the mentioned challenge of service composition.

The core process of our solution consists of three steps. Firstly, to effectively guide the composition process, we identify appropriate mining metrics to establish social service provider network in service oriented cloud environments. Service clients can navigate the network and identify the trustworthy owner of the required services and resources. Secondly, an efficiently discovering approach is proposed to identify a group of collaboration service providers in the constructing social network. In addition, a case study is conducted to illustrate the proposed approach in details. The results indicate the proposed approach can help service consumers discover a set of collaborative providers supporting service composition.

2. Construction of Social Service Provider Network

Socially aware service discovery is proposed to overcome the limitation of the current service composition model. It is critical to collect social information and build social networks for service composition. In this section, we will suggest a new framework for constructing social network of service providers.

Service composition is defined as a meaningful combination of service functions, or called *service type*. The available service candidates can be selected according to their functions. In this article, the term service type is defined as a specific function implemented by web services. Multiple providers often offer the web services with the same service type.

Definition 1 (Service Type). Service type involved in the service composition are denoted in $\{S_i \mid 0 < i \leq n\}$; Services having the same service type are denoted by

$SType(S_i) = \{s_{ij} \mid 1 < j < m_i\}$. A composite service is a set of selected services for each required service type $s_{ij} \in SType(S_i)$.

Services do not work alone, but always collaborate with other services to accomplish a complex project. The collaboration relation refers to the way that two providers may provide different services in a single workflow instance. Collecting execution history is to keep track of past and ongoing workflow executions. Event logs are utilized to store interaction history information for mining service provider networks. From learning the event logs, it can find some useful knowledge to know with whom the service providers have worked in the past. Therefore, in the future, with whom they would collaborate well.

Definition 2 (Event Logs). *An event* represents the state of a service execution. Let *T* be a set of tasks or activities, and *S* be a set of the required services to fulfill *T*. An event is defined as a tuple of $E = \{Task_{id}, Service_{id}\}$, $(t_i, s_i) \in E$ represents that task t_i is accomplished by service s_i. $E = T \times S$ consists of the set of feasible events. *A case*, which is denoted as *C*, is the sequence of possible workflow executions. *An event log* $L \in \mathrm{B}(C)$ consists of a set of cases.

The relations of workflow and service *W* are formalized as $m \times n$ matrix, where *m* is the number and *n* is the number of services in the log: $W = \left[w_{ij}\right] \; for \; 0 \le i \le m, 0 \le i \le \mathrm{n}$. If workflow instance *i* contains service *j*, then $w_{ij}=1$. Each transaction case has recorded the execution information of one instance described by a sequence of events. Only the requirement of mining information is stored in our service event logs.

Service dependency is a linking relationship between two services in a same workflow instance. The detail of service dependency is detailed in definition 3.

Definition 3 (Service Dependency). Let τ be a trace of a service workflow, *L* be a workflow log, and *x* and *y* belong to the set of services *S*; the service dependency can be defined based on the loops between two services as,

$$x \Rightarrow y \text{ , iff } \tau = s_1 ... s_n \, and \, s_i = x, s_j = y (1 \le i < j \le n-1) \qquad (1)$$

Accordingly, the service dependency can be classified into direct successor and indirect successor with two hops, and *n-1* loops. And we can derive service dependency relations *SS* from the matrix *W* : $\mathrm{SS{=}W^T \cdot W}$, where ss_{ij} represents the number of workflow instances where both service *i* and *j* are executed and ss_{ii} is the number of workflow instances where service *i* is invoked. A social service network connects the entire cross organizational services together by social links. A social link represents a relationship between two services. Furthermore, we can get the relations of the services and service providers.

Definition 4 (Service and Provider). Let $\mathrm{P}=\{o_i \mid 0 < i \le l\}$ be a set of service providers, and $\mathrm{S}=\{s_i \mid 0 < i \le n\}$ be a set of *n* web services. Service provider *i* is associated with a set of web services $S(o_i) \subseteq S$. $s_j \in S(o_i)$ means that organization o_i provides web service s_j for a service consumer. The relations of services and providers can be formalized as $n \times l$ matrix: $SP = \left[sp_{ij}\right] \; for \; 0 \le i \le n, 0 \le j \le l$.

Then we derive the relations of the providers and workflows $WP = W \cdot SP = \left[wp_{ij}\right] \; for \; 0 \le i, j \le n$. Similar to service dependence, the relations of the services and service providers can be calculated as $PP = WP^T \cdot WP = \left[pp_{ij}\right] \; for \; 0 \le i, j \le n$, where pp_{ij} is the number of workflow instances where both service provider *i* and *j* have collaborated and pp_{ii} is the number of workflow instances where service provider *i* have participated. Link measure metrics is used to analyze the service provider causality through the sequence relationships of service instance cases. Two subsequent activities in the same case imply the two service providers are causally related. In this context, we can add an arc between the two services provider in the social graph.

Through learning social interaction history of service providers, the knowledge can be extracted to know with whom the service provider collaborated effectively in the past. We use association rules with support or confidence evaluation metrics to estimate service provider relations. The *support* $x \Rightarrow y$ is the fraction of transactions that contain both service providers *x* and *y*. And the *confidence* $c(x \Rightarrow y)$ measures how often a service provider *y* appears in a workflow instance that contains service provider *x*. Therefore, for each service provider, dependency measures are defined as follows.

$$s(x \Rightarrow y) = \frac{\sigma(x \cap y)}{n}$$
$$c(x \Rightarrow y) = \frac{s(x \cap y)}{s(x)} \quad (2)$$

where n is the total number of the workflow instances in event logs, $\sigma(x)$ is the total number of the events with service *x*, $x \cap y$ indicates that service *x* is directly followed by service *y* in one case, and $\sigma(x \cap y)$ is the number of events having the dependency of *x* and *y*.

As the matter of fact, the interactions of services reflect the relationships of the service providers. The relationships of services and corresponding providers are the primary information to construct the service provider social network. The service provider social network can be modeled as a weighted graph.

Definition 5 (Social Service Provider Network). A social provider network reflects the collaborative relationships of service providers; its structure can be described as a directed graph $G = (V, E)$, where: *V* represents a set of nodes, with each node corresponding to linked social service providers; and E is a set of directed edges, with each edge being social link; The weight assigned on each edge is the collaboration cost between the two linked service providers.

A social provider network connects different distributed service providers together by social link. Note that the edge with lower weight values means better interaction between two providers. For instance, if two organizations work together more times in the same workflow, the weight on the edge linking the two providers would be lower assuming they will collaborate more efficiently than two providers who never have collaborated before. This can be utilized to capitalize on the willingness of service providers to collaborate and interact for improving the quality of service composition.

3. Socially Aware Discovery for Service Composition

Due to the development of Internet and social network, an increasing number of service providers are becoming social. It is urgently needed to study how to discover a set of effective collaboration providers in social networks. In this context, we assume that service composition takes place in a complex social network.

The success of social aware service composition depends on how effectively all involved service providers collaborate and communicate with each other. Social communication cost is defined to calculate collaboration efficiency for socially aware service composition. It is determined by the weights of the shortest path connecting both providers *i* and *j* in the social graph. Social communication cost is defined in the following ways.

Definition 6 (Collaboration Cost). Given two nodes $i, j \in V$, the distance $dist(i, j)$ is the sum of weight of edge weights along the shortest $path(i, j)$ which is the sequence of nodes along the shortest path; the distance between a node i and a set of nodes V' is defined as $dist(i,V') = min_{j \in V'} dist(i, j)$; and the communication cost of $V' \subset V$ is defined as the sum of edge weights in minimum spanning tree of the specified subgraph *SG*, denote as $totalCost(V')$.

In a socially aware service composition problem, the expected service providers must satisfy: (i) service providers' alliance can provide all the required services to complete the given business goals; (ii) all service providers must be capable of collaborating effectively. In other words, it is essential to find a set of service providers to accomplish the required tasks and the total collaborating cost is as low as possible. Given a social graph G=<V, E>, and the required service types denoted as $ST = \{st_i \mid 1 \le i \le S_{num}\}$; the service composition problem in social provider network is to find a subnet work SG=<V, E> of the social graph. Formally, it can be described as follows.

$$\begin{aligned} &Minimize: totalCost(V'). \\ &Subject\ to: cover(V', ST) = 1. \end{aligned} \tag{3}$$

To solve the problem of service composition in social network, we present an effective discovery approach inspired in the area of team formation. The Enhanced-Steiner algorithm is extended and adopted in our proposed composition approach. It was firstly proposed solving original group formation in the social network [9]. The Enhanced-Steiner algorithm suffers from poor efficiency when the social network has too many members and relationships. Group graph based service discovery algorithm needs to be able to improve time efficiency because it reduces the search space.

The example of the group graph is shown in Figure 1. Artificially additional nodes are service type nodes. Bold lines represent previous collaborations between two service providers, while dotted lines are the new added edges from each service type node to the corresponding service providers. A group graph can be utilized to simplify discovery calculations. Group graph can be formulated as follows.

Definition 7 (Group graph). A group graph $H = (V_H, E_H)$ is a weighted graph derived from the original network $G = (V, E)$. The required service types are defined as group nodes V_H; E_H is a set of group links between two nodes grp_{Si} and grp_{Sj}. The weight of E_H can be calculated as $weighted(e_H) = min\{dist(u, v)\}$, where $u \in V(grp_{Si})$ and $v \in V(grp_{Sj})$.

In order to efficiently guide the composition process, the proposed composition approach aggregates the social network into group graph structure. The first step of our algorithm is remodeling social graph based on grouping graph, which adds the providers possessing the same service type into one group. This step is adding additional nodes and edges for the constructing graph. A new additional node *Sj* is created in the same group. The added vertex is called service type node. Such the new functional node is connected to every node of service provider who offers services satisfied the service type. The distance between task node *Sj* and associated service provider node *Oij* is assigned to be *D*, where *D* is a large real number.

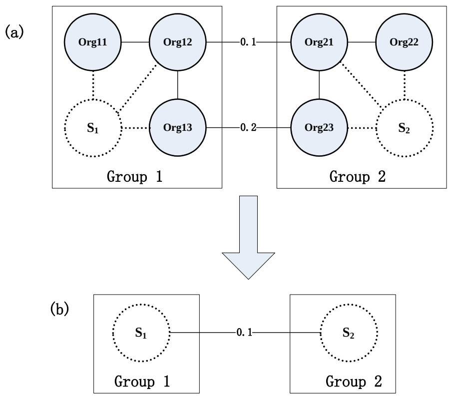

Figure 2. (a) Enhanced Graph (b) The Simplified Group Graph

As shown in Figure 2, the original social graph is simplified as two node graph. Observably, the structure of the original graph is simplified. It reduces the search space efficiently. Based on the mentioned definition and analysis, socially aware service composition is stated in Algorithm 1. The modified *Steiner-Tree* algorithm [11] is employed to find the minimum cost tree in the group graph. The proposed algorithm iteratively identifies and adds the selected service into the composition solution.

Algorithm 1. Socially Aware Service Composition

Input: A given service composition with required service types $SType = (st_i)$, service event logs, relationship of providers and services.
Output: A group of collaborative providers V' and $SG = <V', E'>$ for service composition

building dependency matrix of service providers
constructing the social service provider network: $G(V,E)$
$H = (V_H, E_H) \Leftarrow$ group graph G
SV'.add(v_{init})*, where v is a random service type vertex from SType*
for each v **in** V_H **do**
  if v not in SV' **then**
    If dist(v, SV') <= min **then**
      SV'.add(v)
   end if
    // search in enhanced graph
    for each provider vertex in $path(v^*, SV')$
      selectedService = getServices(*provider*)
      wsolution.add(*selectedService*)
    endfor
  end if
end for
return *wsolution*

The socially aware composition algorithm is inspired by the *Enhanced-Steiner* algorithm. In each step, a single service provider v^*, which has the minimum distance to all nodes in *SV'*, is added. And the service, supplied by the providers on the path between v^* and *SV'*, is selected in the composition solution. The time complexity of the proposed algorithm includes three parts: (a) the time complexity of the grouping graph is $O(V_{num})$, where V is the number of the provider candidates; (b) the running time of $O(V_H \times E_H)$; (c) the computation of the shortest paths between the two group nodes is $O(V_H \times E)$.

4. A Case Study

In this section, a case study is used to illustrate the detail of the proposed approach. Firstly, social graph of service providers is constructed in formalism way step by step according to the definitions of social graph construction. Secondly, social network analysis technologies and tools [12] is applied to quantify the structure and the metrics of social graph. Finally, we explain the proposed algorithm of social aware service composition.

4.1 Building Social Network of Service Providers

Table1. Service Types and Services

Service ID	Available Services
S_1	s_{11} s_{12} s_{13}
S_2	s_{21} s_{22} s_{23}
S_3	s_{31} s_{32} s_{33}
S_4	s_{41} s_{42}
S_5	s_{51} s_{52} s_{53}

Table 2. Services and Their Providers

Provider ID	Service ID
O_1	s_{11} s_{23}
O_2	s_{12} s_{s2}
O_3	s_{31}
O_4	s_{32} s_{41}
O_5	s_{33}
O_6	s_{13}
O_7	s_{42} s_{51}
O_8	s_{52}
O_9	s_{21} s_{53}

Table 3.Service Workflow Logs

Instance ID	Service Event Stream
Wid_1	s_{11} s_{21} s_{31}
Wid_2	s_{31} s_{51}
Wid_3	s_{21} s_{32} s_{41} s_{51}
Wid_4	s_{11} s_{22} s_{33}
Wid_5	s_{31} $s_{41}s_{51}$
Wid_6	s_{21} s_{31}

Service composition is combined with a set of *service types*. The available service candidates can be selected according to their functions. The specific functions are implemented by web services, as shown in Table 1.

A composite workflow is composed of a set of web services which are provided by different organizations. An organization often provides various web services. Table 2 shows the information of the services supplied by the organizations. The relation matrix *SP* of services and their providers can be formulated based on *Definition 4*.

According to the example data shown in Table 1, Table 2, and Table 3, the relations matrix of workflow instances and service providers can be calculated in the aforementioned formalism way. The detail calculation is demonstrated as the following ways.

Let *WP* be the relation matrix of workflow instance and service provider, and *PP* be the relation matrix of service providers then

$$WP = WS \cdot SP = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad PP = WP^{T} \cdot WP = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Furthermore, based on Equation 1, the matrix of social relations (*MSR*) can be formalized from the matrix *PP*, as shown in the following matrix.

$$MSR = \begin{pmatrix} 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 1 & 0.25 & 0 & 0.5 & 0 & 0.5 \\ 0.17 & 0.17 & 0.17 & 0 & 0 & 0 & 0.5 & 0 & 0.33 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.67 & 0 & 0 & 0 & 0 & 0 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.33 & 0 & 0.67 & 0.67 & 0 & 0 & 0.33 & 0 & 0 \end{pmatrix}$$

4.2 Analyzing the Social Service Network

To develop compositions, service engineers need to identify a group of collaboration owners of the required services and resources. Through the constructing graph, the proposed service composition social link analysis has been applied successfully to a wide range of Web 2.0 applications. It refers to the analysis of the relationships among human actors or organizations in a social network. Social link analysis helps researchers identify entities and their degrees of influence in a social network. A drawing of a graph can be utilized to demonstrate the important social features in an efficient way. In this section, social network analysis techniques are applied to study the structure and metrics of the networks.

Linked social service providers are constructed upon the knowledge extracted from event logs. The relations of workflow and service are shown in Figure 3(a). Blue circles represent workflows, red squares represent services. The *relation of workflows and service providers* are derived from the relations of workflow and service, as shown in Figure 3(b). Through these networks, we can know how workflows utilize services; and how service providers have ever collaborated with one another. Such knowledge will facilitate service composition reuse.

To clearly analyze and understand the proposed model, we build service provider social network according to service provider dependency matrix. The network of service providers is given in Figure 4(a). Circles represent service provider nodes; squares represent the service type nodes. Different colors mean different service types.

Observably, the social network is aggregated into group graph structure. Providers supplying the same service type are connected into one group.

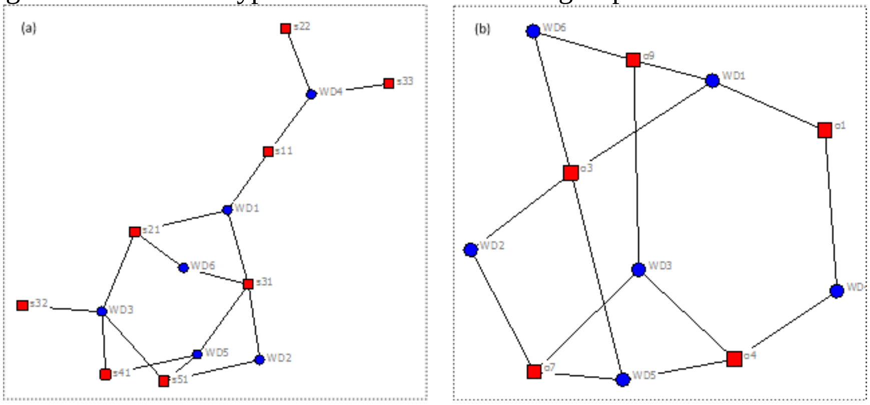

Figure 3. (a) Relations of Workflows and Services (b) Relations of Workflows and Providers

In the graph theory and network analysis, *centrality measurement* evaluates the relative importance of a vertex within a graph. It could be employed to identify the highly used services and service providers. *Degree centrality* is defined as the number of links incident upon a node. And betweenness is a centrality measure of a vertex within a graph. *Betweenness centrality* quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It was presented as a measure for quantifying the control of a human on the communication between other humans in a social network. In this section, betweenness centrality is used to evaluate communication-based importance of the service providers, as shown in Figure 4(b). The larger of the node represents the more significant in the graph. The algorithm is coded in Java. It allows service consumer to start browsing in one service provider and then navigate along social path to find the minimum cost tree that spans all the providers of the required services.

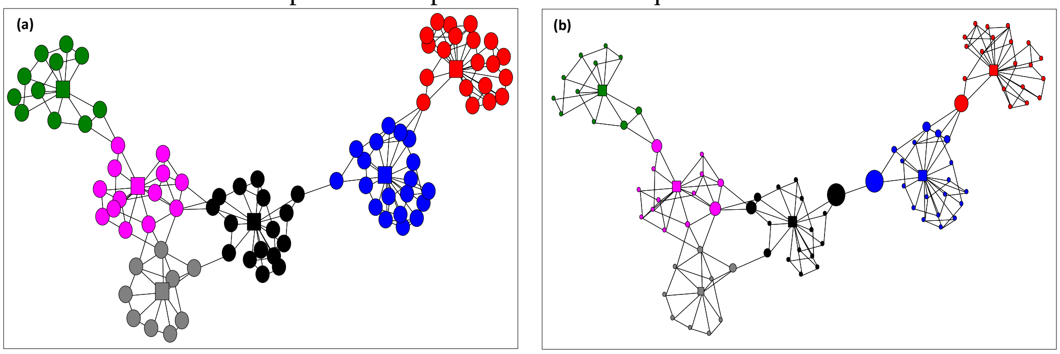

Figure 4. (a) The Social Service Provider Network (b) Centrality Measurement of the Social Network

4.3 Discovering Providers in the Social Graph

To develop service compositions, service engineers need to identify a group of collaboration owners of the required services. Through the constructing social graph, the proposed composing method is used to discover the required services. The algorithm is coded in Java. It allows service consumers to start browsing in one service provider and then navigate along social path to find the minimum cost tree that spans all the providers of the required services.

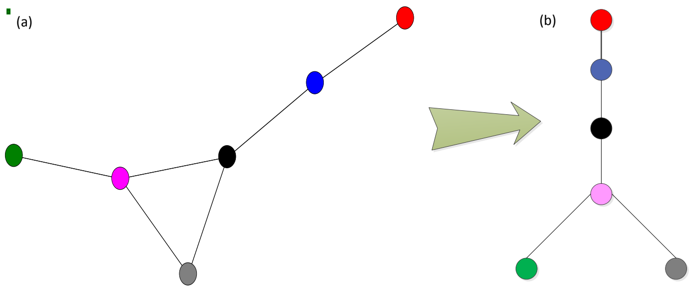

Figure 5. (a) Simplified Structure of the Original Graph (b) the Minimum Spanning Tree

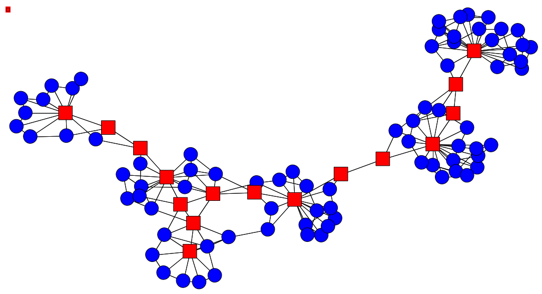

Figure 6. The Search Results in the Social Provider Network

According to algorithm 1, it consists of three steps to find a set of the required service providers. Firstly, the source graph shown in Figure 4(a) is formalized as group aggregation spanning tree with more simplified structure, as shown in Figure 5(a). Secondly, we can gain the minimum spanning tree; the root of the tree is S_1; and the printing sequence of the tree is $(S_1, S_2)(S_2, S_3)(S_5, S_4)(S_3, S_5)(S_5, S_6)$, as shown in Figure 5(b). Finally, the discovery process searches the required service providers through the social minimum path between the two service type nodes. The search results are presented in figure 6. The red square nodes represent the search trace and the selected nodes for service composition.

5. Conclusion

In this paper, we have presented a socially aware discovery approach for service composition and an empirical case study of the proposed approach. Firstly, process analysis technique is employed to discover the hidden knowledge of the service event logs. Secondly, our study utilizes the mining relative knowledge to construct a social graph for service composition. And then, based on the analysis method of group graph, the original social graph is remodeled as a simpler graph. Finally, through the constructing social networks, a social discovery method is proposed to identify a group of effective collaborative service providers for satisfying the business requirements. The proposed approach is chosen because it is capable of efficiently addressing the objective of service composition. In the conducted case study, the performed results indicate our methodology can effectively discover service providers' alliance to achieve the given business goals; and all the selected service providers are capable of collaborating effectively.

In the complex social network, the number of alternative candidates may be very large; and performing an exhaustive search to find service providers may incur substantial computational cost. As future work, we will try to develop an effective approach of service composition in a complex social environment.

References

[1] S. Dustdar and W. Schreiner, "A survey on web services composition", International journal of web and grid services, vol. 1, no.1, **(2005)**, pp.1-30.
[2] M. Bichier and K. J. Lin, "Service-oriented computing", Computer, vol. 39, no. 3, **(2006)**, pp.99-101.
[3] I. Foster, "Service-oriented science", Science, vol. 308, no. 5723, **(2005)**, pp.814-817.
[4] R. Eshuis, A. Norta, O. Kopp and E. Pitkanen, "Service Outsourcing with Process Views", IEEE Transactions on Services Computing, **(2014)**.
[5] D. Bianchini, C. Cappiello, V. De Antonellis and B. Pernici, "Service identification in inter-organizational process design", IEEE Transaction on Service Computing, **(2013)**.
[6] Z. Maamar, P. Santos, L. Wives, Y. Badr, N. Faci and J. P. M de Oliveira, "Using social networks for web services discovery", IEEE Internet Computing, vol.15, no.4, **(2011)**, pp. 48-54.
[7] W. M. Van Der Aalst, H. A. Reijers and M. Song, "Discovering Social Networks from Event Logs", Computer Supported Cooperative Work, vol. 14, no. 6, **(2005)**, pp. 549-593.
[8] K. Gaaloul, S. Bhiri, A. Haller and M. Hauswirth, "Log-based transactional workflow mining", Distributed and Parallel Databases, vol. 25, no. 3, **(2009)**, pp.193-240.
[9] C. T. Li, M. K. Shan and S. D. Lin, "On team formation with expertise query in collaborative social networks", Knowledge and Information Systems, **(2013)**, pp.1-23.
[10] T. Lappas, L. Liu and E. Terzi, "Finding a team of experts in social networks", Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, **(2009)**; Paris, France.
[11] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs", Mathematica Japonica, vol. 24, no. 6, **(1980)**, pp. 573-577.
[12] S. P. Borgatti, "NetDraw: Graph Visualization Software", Harvard: Analytic Technologies, **(2002)**.

Author

Yong Sun received the master's degree in computer science from China University of Geosciences (Wuhan) in 2007. His fields of research are software engineering, cooperative computing, service computing, grid and cloud computing based workflow management and intelligent information systems.