1-1-2011

Composite Web Services Formation Using a Social Network of Web Services: A Preliminary Investigation

Said Elnaffar
United Arab Emirates University

Zakaria Maamar
Zayed University

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 466–471

Procedia Computer Science

The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)

Composite Web Services Formation Using a Social Network of Web Services: A Preliminary Investigation

Said Elnaffar[a]* and Zakaria Maamar[b]

[a]*Faculty of IT, UAE University, Al Ain,UAE*
[b]*College of IT, Zayed University, Dubai,UAE*

Abstract

A composite Web service, such as *arranging a travel to a conference*, can be viewed as a project team that is comprised of several members who want to collaborate in order to accomplish a specific goal. In this work, we propose an algorithm for searching a social network of Web services in order to select an effective set of Web services that can collaborate in order to attain the main goal of the composite Web service at hand. Ultimately, the selection algorithm takes into consideration the set of functions that each Web services member possesses, e.g., hotel booking, car rental, etc., and the communication cost among these members. Our goal is to secure all the functions required by the composite Web service yet minimizing the communication overhead. The preliminary design of the experiments and the future work are described.

Keywords: composite web services; team formation; searching social network

1. Introduction

Web services discovery is critical to fulfilling users' requests. A plethora of Web services exist on the Internet which makes identifying the right ones not always a straightforward task. Additionally, independent providers continue to deliver almost the same set of Web services with different non-functional properties, which further increases the complexity of the discovery process. Nevertheless, these Web services interact with each other incessantly in order to compose bigger Web services that can eventually form a full-fledged IT system or product. The interactions among Web services construct a network of relationships leading to a social network of Web services [13][14].

A social network of Web services can be viewed as a graph (G) whose nodes are Web services connected to each other by communication edges (E). An undirected edge between two Web services means that they potentially can communicate with each other. The weight of the edge represents the communication cost (latency). Each Web service in the social network has one function or more. Each function is tagged by a QoS grade that reflects the historical performance quality of this function. Without losing generality, a grade can be low, medium, or high. The historical performance of a each Web service can be maintained by different ways such as a centralized entity within the social network or by a service broker.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.
doi:10.1016/j.procs.2011.07.060

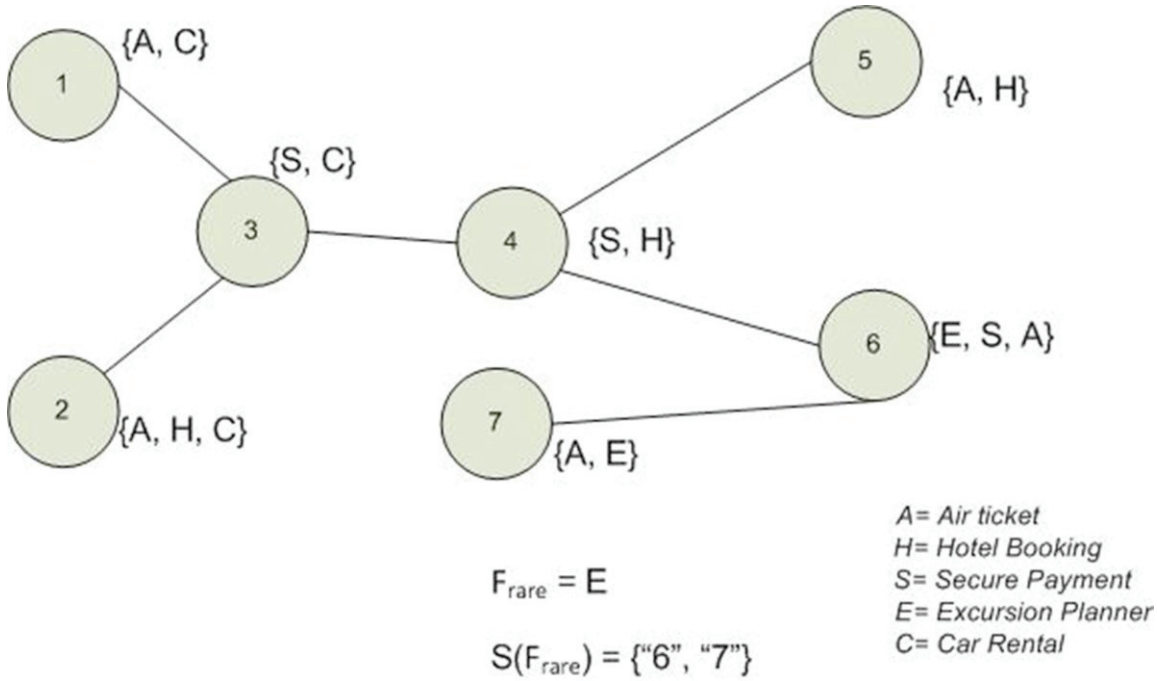

Fig. 1: An example of a Web services social network

Depending on the need of the composite Web service, it should specify the functions needed for its implementation and the minimum QoS grade of each function. Concurrently, we wish to have these sub-WS to communicate with each other effectively by minimizing the communication cost. In this work we study the problem of extract a set of Web services that 1) possess the required functions with minimum grades of QoS and 2) can communicate efficiently in order to undertake specific task. This set of Web services constitutes a connected sub-graph *G'* where $G' \subseteq G$.

Fig. 1 depicts an example of a social network of Web services. Each node represents a Web service and each edge indicates a communication channel between them. Web service "2", for example, supports three functions: Air Ticket Reservation, Hotel Booking, and Car Rental. Let's imagine having *ScholarCompanion* as a hypothetical composite Web service that is supposed to take care of all details germane to a researcher's trip to a conference. Essentially, this Web service has to handle the following list of functions with their associated QoS grades:

Table 1. *An example of functions needed by a composite Web service.*

Function	**Min. QoS Grade Required**
Air ticket	Medium
Hotel booking	Low
Car rental	Medium
Secure Payments	High
Excursion Planner	Low

The QoS grades are implementation-dependent and can be assigned, for example, by a centralized rating entity such as the service broker who monitors the performance of each Web service upon completion and collects feedback from its consumers. We are looking to find a set of Web services that can handle the above functions with the minimum grade indicated and minimum cost of communication among them.

The rest of the paper is organized as follows: Section 2 gives a brief description of the background of our work. Section 3 introduces the definitions and terminology that are used throughout the paper. Section 4 explains our algorithm in detail, and Section 5 gives some details of the current implementation and sketches future work.

2. Related Work

The social networks of WSs are different from the classical type of social networks of people (e.g., LinkedIn, Plaxo, Facebook) [13]. The latter is based on the absolute collaboration and mutual assistance between their members (i.e., no competition). Contrarily, in WSs social networks, members are mainly competing as each Web service wishes to be (i) a part of compositions, (ii) a replacement for a faulty one, and (iii) nominated as an add-on Web service to augment the currently invoked one (e.g., a Web service *to arrange for conference participation*, and an add-on Web service is *to arrange for an excursion*).

Our work in the context of social networks of Web services is motivated by the need for finding a set of collaborative Web services that can fulfil the functions required by a composite Web service while they can communicate efficiently. This work is inspired by a number of efforts done in the classic social networks of people. For example, in the context of project management, researchers used Integer Linear Programming techniques to solve the problem of team formation [4, 5, 6, 8]. For the purpose of optimizing the selection based merely on the human skills, they used simulated annealing [4], branch-and-cut method [6], and genetic algorithms [8].

Askin and Fitzpatrick [3] focused on forming qualified teams with multi-functional skill requirements by computing the effectiveness using Kolbe Conative Index [12]. Gaston and des Jardins [9] investigated how team formation occurs and the influence of that formation on the performance of the project at hand. They also suggested some approaches for agent-organized networks. H. Wi et al. [10] introduced a methodology by which a person's skill and knowledge are assessed. Subsequently, teams get formed based on this assessment.

Most of the above studies overlooked the connections that existed among people despite their existence in a human-based social network. We believe, however, that taking the efficiency of communication between human being is a key factor when forming a work team. On the other hand, some other studies [1] paid attention to the graph structure but ignored the skills of individuals needed to complete the job. L. Backstrom et al. [7] delve into many aspects of large social networks especially the graph structure and provides some insights of why a person may move from one community to another.

In this work, we are trying to combine the two criteria, functionality and inter-relationships, in order to come up with an efficient set of Web services that constitute the bigger composite Web service.

3. Problem Definition

Formally, let us assume that our social network of Web services consists of n Web services $SN = \{w_1, w_2, \dots, w_n\}$ that comprise a graph *G(SN, E)* where *E* is a set of edges connecting the Web services. We denote the universe of the Web service functions by $F = \{f_1, f_2, \dots, f_n\}$. Each Web service w_x has a set of functions F_x, where $F_x \subseteq F$.

A composite Web service may require a set of functions, *T*, where $T \subseteq F$. So, if $f_x \in T$, then the function f_x is required and must be supported by at least one of the Web services finally selected. We call the set of Web services that offer the function f_x, with the minimum QoS grade specified, the support of that function, denoted by $S(f_x)$. According to the social network of Fig. 1, F = {A, H, C, S, E}, S(A) = {"3", "4", "6"}, $|S(A)| = 3$.

We assume that Web services communicate via the edges, *E*, of the undirected graph *G(SN, E)*. Each edge is weighted to reflect the communication cost between the pair of Web services. If there is no edge between two Web

services, we choose to assign a very heavy weight to that edge to reflect such a situation. $P(w_j, w_k)$ denotes the path (node sequence) between w_j and w_k, and $Comm(w_j, w_k)$ refers to the communication cost of the shortest path between the two Web services measured using Dijkstra's Shortest Path. Our goal is to find the sub-graph $G'(SN') \subseteq G(SN)$ using our algorithm *Composite Web Service Builder*, which is explained next.

4. The Algorithm

In essence, we are working on an optimization problem where we wish to select Web services that fulfil the required functions while they 1) are similar to each other functional-wise, and 2) communicate with each other at the minimum cost. The functional similarity constraint adds reliability to the operation of the composite Web service in case one of the constituent Web services fails (i.e., better fault-tolerance). The communication cost is computed by measuring the social network diameter, which is the longest shortest path between any two nodes. The designer of the composite Web service will be able to determine how much he would like to rely on either of these measures using a bias parameter which will be introduced shortly.

Fig. 2 sketches *the composite Web service builder* algorithm. As inputs, we have a graph of Web services and a set of functions *T* required by the prospective composite Web service. We initially calculate the support, $S(f_x)$, for each function in *T*. Then we find the one with the least cardinality and label its function as f_{rare}. Therefore, f_{rare} is the function that is supported by the least number of Web services. In our running example (Fig. 1), f_{rare} is the function *E* (Excursion Planner) which is supported by the Web services "6" and "7".

In step 2, we traverse the network from w_{rare} to w_i moving along the edges in order to calculate the distance of each path, and eventually select the Web service w_i that gives the minimum distance. The distance is computed based on the combination of two components: similarity in functionality and ease of communication. Both properties are desired for our optimization problem.

With respect to the communication cost between two Web services, each edge in the graph of the social network has a pre-computed/experienced value that reflects the QoS expected from the Web service. Concerning the similarity of functions, and upon securing the mandatory set of functions *T* required by the composite Web service, it is desirable to have the underlying Web services similar with respect to the functions they provide as this should increase the reliability of the composite Web service in case of failures. To measure the similarity of functions, we use Jaccard's pairwise distance [15].Therefore, the similarity between Web service w_j and w_k can be expressed as:

$$Sim(j,k) = \frac{|F_j \cap F_k|}{|F_j \cup F_k|}$$

Accordingly, we can now define the path distance as follows:

$$D = (1-\alpha)(1 - Similarity) + \alpha(communication\ cost\ over\ the\ path), D \in [0,1]$$

Notice that in [1], the authors considered one component only which is the similarity of the skills among the workers in order to form a team that have similar interests. The communication cost was not factored in. Nevertheless, we argue that the communication cost among people should have been also captured in their model by assessing how easy for two team members to communicate and get along with each other.

The α parameter in the above equation gives the designer of the composite Web service the ability to be biased toward one of the components more than the other. The communication cost from w_{rare} to w_i, $Comm(w_{rare}, w_i)$, is the physical measurable distance of the path between these two Web services, which is calculated using Dijkstra's shortest path algorithm. Therefore, we further refine the above equation as follows:

Algorithm: Composite Web Services Builder
INPUT: Graph *G(SN, E)*; Task T
OUTPUT: $G'(SN') \subseteq G(SN), SN' \subseteq SN$

1. $\forall f_x \in T$
   a. Compute $S(f_x)$
   b. Find f_{rare} such that $|S(f_{rare})|$ is the minimum of $\forall S(f_x)$
2. $\forall w_{rare} \in S(f_{rare})$
   a. $\forall f_x \in T$
      i. $\forall w_i \in S(f_x)$
         Calculate all path distances $D(w_{rare}, w_i)$
      ii. Find the diameter of $S(f_x)$ // longest shortest path between 2 Web services.
   b. Determine Web service w_{rare} with the shortest diameter
3. Add to *SN'* the Web service from each $S(f_x)$ that minimizes the distance function
4. Draw the graph *G'(SN')* starting from Web service w_{rare}.

Fig. 2: Composite Web Service Builder Algorithm

$$D(w_{rare}, w_i) = (1-\alpha)(1 - Sim(w_{rare}, w_i)) + \alpha\left(\frac{Comm(w_{rare}, w_i)}{n}\right), D \in [0,1]$$

where n = number of Web services along the path from w_{rare} to w_i. This is to normalize the communication cost and keep the value of $D \in [0,1]$.

5. Implementation and Future Work

Our implementation is a work in progress. We are presently implementing the proposed algorithm using the Java Universal Network/Graph (JUNG) where we generate a graph of 2000 nodes (Web services) connected by almost 4000 edges. The network is constructed using the Eppsein power law graph generator [11], where node degrees follow the power law distribution. Each Web service is assigned a function from a set of 100 different functions. Using a uniform distribution between [0, 1], each function is assigned a QoS grade that represents its historical performance. We read the various functions required by the composite Web service from an XML file that has the functions and the minimum QoS grade.

Tentatively, we would like to observe the number of the Web services selected in relation with the number of functions required by the composite Web service. The quality of the selected Web services is assessed in terms of the size of the generated sub-graph. Furthermore, we plan to try other techniques for optimizing the communication cost such as the Minimum Spanning Tree (MST) and contrast their performance without the methodology presented in this paper.

References

1. T. Lappas, K. Liu, E. Terzi, Finding a Team of Experts in Social Networks. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 28 -July 01, 2009, Paris, France.
2. L. Lai and E. Turban, Groups formation and operations in the Web 2.0 environment and social networks. Group Decision and Negotiation, 17 (5) 387-403, 2008.
3. E. L. Fitzpatrick and R. G. Askin. Forming effective worker teams with multi functional skill requirements. Comput. Ind. Eng., 48(3):593-608, 2005.
4. A. Baykasoglu, T. Dereli, and S. Das. Project team selection using fuzzy optimization approach. Cybern. Syst., 38(2):155-185, 2007.
5. A. Zzkarian and A. Kusiak. Forming teams: an analytical approach. IIE Transactions, 31:85-97, 2004.
6. S.-J. Chen and L. Lin. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. IEEE Transactions on Engineering Management, 51(2):111-124, 2004.
7. L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In KDD '06:Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 44-54, New York, NY, USA, 2006. ACM.
8. H. Wi, S. Oh, J. Mun, and M. Jung. A team formation model based on knowledge and collaboration. Expert Syst. Appl., 36(5):9121-9134, 2009.
9. M. E. Gaston and M. des Jardins, "AgentOrganized Networks for Dynamic Team Formation," in Autonomous Agents and MultiAgent Systems (AAMAS) Utrecht, Netherlands, 2005, pp. 230-237.
10. Hyeongon Wi , Seungjin Oh , Jungtae Mun , Mooyoung Jung, A team formation model based on knowledge and collaboration, Expert Systems with Applications: An International Journal, v.36 n.5, p.9121-9134, July, 2009.
11. Eppstein, D. and Wang J., “A steady state model for graph power law”, 2nd international workshop on Web Dynamics, Honolulu, Hawaii 2002.
12. Kolbe, Kathy, *The Conative Connection,* Addison-Wesley, Reading, MA, 1990.
13. Zakaria Maamar, Leandro Krug Wives, Youakim Badr, and Said Elnaffar, "Even Web Services Can Socialize: A New Service-Oriented Social Networking Model," in the proceedings of the IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), Barcelona, Spain, November 4 - 6, 2009.
14. Zakaria Maamar, Hakim, Hacid, and Michael N., Huhns, “Why Web Services Need Social Networks", IEEE Internet Computing, Vol. 15, No. 2, March/April 2011.
15. Jaccard, Paul (1901), "Étude comparative de la distribution florale dans une portion des Alpes et des Jura", Bulletin de la Société Vaudoise des Sciences Naturelles 37: 547–579.