

Towards a Social Network Based Approach for Services Composition

Abderrahmane Maaradji, Hakim Hacid, and Johann Daigremont
Alcatel-Lucent Bell Labs France
Centre de Villarceaux Route de Villejust, 91620 Nozay, France
Email: {firstname.lastname}@alcatel-lucent.com

Noël Crespi
Institut Telecom, Telecom SudParis
9 Rue Charles Fourier, 91000 Evry, France
Email: noel.crespi@it-sudparis.eu

Abstract—With the emergence of Web 2.0 and the related technologies, composing services has left the traditional frontiers of enterprises. In fact, end-users need to use a certain kind of composition in different situations since Web 2.0 has brought a set of technologies making it easy to create or collaborate on new services or use others' services, e.g., mashups. On the other hand, users participate in different communities and social networks to share common interests and find expertise offered by others. Even if Web services composition tools like mashups include a community dimension helping the service creation process (tagging, rating, ...), they completely ignore this social dimension at the composition level. Our approach is to bring the generated knowledge from interactions between users and services (and by extension from social environments) to enhance services composition. This paper reviews some related concepts and work and then introduces a first view of our framework, named *Social Composer* (SoCo), aiming at handling this issue from a social networking perspective. *SoCo* provides dynamic recommendations for services discovery and selection based on the users' interactions and a social network implicitly built from the interactions between users and services, and the different services compositions operated in the user's social network as well as the global social network.

I. INTRODUCTION

Creating value-added services by reusing existing ones, known as services composition, has been a key issue service science and heavily investigated from both industrial and academic perspectives [1][2]. This is motivated by the fact that numerous services need to be produced quickly because of the growing demand of customers. Creating services from scratch, to satisfy the increasing demand, needs too many programmers, costs too much, and takes too long to reach the market. Another important goal of service composition is to provide the ability to customize services according to end-user's preferences. This approach facilitate provide end-users by personalized and user-centric services. Services composition has been addressed in many projects related to web services. Initially, Web services composition was dedicated for expert users like programmers. However, with the emergence of the Web 2.0, it is becoming more important to make the composition process much more end-user oriented. In fact, the Web 2.0 has brought a set of different technologies dedicated for end-users who do not have a strong expertise in programming. It became then very easy for such users to publish or annotate resources, stay in touch with their

social relatives, and even combine services using the growing paradigm of mashups [3], [4].

Web services composition can be performed either (i) manually, (ii) automatically, or (iii) semi-automatically. For the first approach, where services are entirely composed by hand in some cases using pure formal languages as SDL [5], end-users need a high programming technical level, which make its use limited. The second approach, which aims to automatically build composite services that match user context or request, faces to indecision problem that need to involve the user. The last approach aims at providing end-users with an enhanced service creation environment. This environment offers support for automated processing of some composition tasks where the end-user operates in a more or less involved manner. Semi-automatic composition, somehow, comes to resolve the situation since significant efforts aim to consolidate this approach by addressing particular issues, for instance the difficulty of selecting a relevant service among the many available ones. We are particularly interested in this approach as we will describe in detail later.

Thus, the semi-automatic approach has the main advantage of making the user participate in the composition process while keeping his overhead as small as possible by exploiting information generated by, e.g., the user or his community. This is specially true in an enterprise context where a community knowledge is used. However, with the emergence of social networks and collaborative environments, composing services has left the frontiers of enterprises and end-users in different situations need to apply a certain kind of composition. Thus, we believe that there is potential to reinforce services composition in such new conditions. A current evolution of semi-automatic composition is what is now commonly called mashups. This latter evolution translates the emergence of Web 2.0 and more specifically the aspect of user generated content (UGC). So, how can we combine social networks and mashups to provide a socially-aware mashup creation environment for end-users?

The rest of this paper is organized as follows: The next section discusses some studies in the area of semi-automatic services composition and the Web 2.0. Section 3 introduces a first attempt of the an ongoing framework aiming at handling this issue. Section 4 discusses an illustrative use-case. We conclude and discuss the different issues in Section 5.

II. SEMI-AUTOMATIC WEB SERVICES COMPOSITION AND WEB 2.0

A. Semi-automatic Web services composition

Semi-automatic Web services composition has taken several forms evolving over time from simple graphical tools to semantic-based tools. As introduced below, mashups are an instantiation of this concept. This has helped the emergence of a multitude of methods for semi-automatic Web services composition which we have categorized into three major categories: (i) *single end-user oriented*, (ii) *domain or community oriented*, and (iii) *social network oriented*.

1) *Single end-user composition*: This category builds a profile of the user, involving him in the non-decision stages, and/or provides tools and interfaces to facilitate the composition of applications. In this approach we can find all BPEL-tools, user-driven composition, mashups, etc. For instance, Law [6] introduces a system called *Koala* (currently Co-Script) characterized by a “side bar” in the user’s browser which learns user’s behavior when processing a web page and transform this behavior into a series of actions.

2) *Community composition*: This approach aims at considering the knowledge produced in communities or in a specific domain, e.g., enterprises. A community can be involved in the process of composition in two ways: (i) in the pre-composition process at the discovery and selection levels, or at post-composition process by annotating, ranking, and rating services. (ii) By Extracting the existing knowledge in this community or domain with a well-defined process which may enable the definition of a set of rules and building a recommendation system in the composition process. Chen et al. [7] provide an interesting introduction to domain-knowledge for services composition. Semantic processing of extracted knowledge based on ontologies which enables a “sophisticated” service discovery [8][9].

3) *Social Networks composition*: The third category of semi-automatic services composition is social networks based. A social network can not be considered in the community approach (the above paragraph) because it describes specific structures. The major difference is that a community describes a gathering of individuals around a common topic of interest, generating communities specialized in particular areas (what justifies this approach). On the other hand, the knowledge available in social networks can be richer than the one of specific communities. Social network describes friends’ networks constructed on the basis of specific interests for each relationship in the network. In other words, the knowledge defined in a social network can not be processed as community knowledge. Therefore, an approach should be established for this particular case. Notice also that mashups built by end-users address specific situations which may eventually fit our problem.

B. Services composition and Web2.0

Web 2.0 harnesses the Web in a more interactive and collaborative manner, emphasizing peers’ social interaction

and collective intelligence, and presents new opportunities for leveraging the Web and engaging its users more effectively [10]. Within the last three or four years, Web 2.0, ignited by successful Web 2.0 based social applications such as MySpace, Facebook, etc. has started to become an interesting source of knowledge to the services composition research community.

Thus, several works have been launched to exploit the knowledge of the masses in order to improve the composition process by considering either social networks or collaborative environments. An example of such work is [11] which proposes a system exploiting collaborative environments to compose skills for performing specific tasks.

On the other hand, mashups are a new emerging paradigm of Web 2.0 enabling end-users to easily create web-based applications and services that address their specific needs and interests. Several IT and Web players offer mashups creation Web environment such as: Microsoft Popfly¹(currently closed), Yahoo pipes², and Open mashups Studio³. Mashups are also used for services composition. Many research studies has been conducted on this topic analyzed this paradigm from several point-of-view. Yu et al. [4] mainly highlight the mashup environment technical aspects from a services providers’ viewpoint. [4] introduces a methodology to be considered for a mashup creation system, and [3] points the internal mashup data models which can be either graph-based or object-based and highlight the need to consider internal and external data integration aspects.

From the end-user point of view, Grammel and Storey [12] investigate tools and environments for creating mashups as “an end user driven recombination of web-based data and functionality”. This analysis highlights needs for considering communities and in mashups tools design.

The difference between a community and a social networks is that the first covers a network of specific interests and the second covers a network of friends or colleagues (eventually with specific interests). After a detailed analysis of social properties, it is interesting to note that none of the selected mashup makers offer social networking features at all. Auhtors in [13] show that the so called existing social networks properties are just community-related and name it “mass collaboration” features.

There is a real need to exploit social networks, i.e., the knowledge generated by social networks, in the composition process. Their use is slowly becoming a reality considering the different attempts that follow this direction. However, until now, there is no particular distinguished work in this area which clearly uses this kind of knowledge. Thus, there is a need for more investment in this direction and the work we are performing fits with this area of research. The next section discusses our ongoing attempt to support social networks in services composition. Our objective at this stage is to show which information can be extracted from social networks in order to be exploited in a composition task.

¹<http://www.popfly.com/>

²<http://pipes.yahoo.com/>

³<http://www.openmashups.org/>

III. TOWARDS SOCIAL-AWARENESS WEB SERVICES COMPOSITION

In Web 2.0, people can create, use, and share services in communities and social networks. These services can be simple web services or more sophisticated services as mashups. The question we are aiming to answer is: *how to leverage social interactions in a way to enable and facilitate composite services creation?*

Regarding this problem statement and the related work, we propose a general framework for services composition based on knowledge retrieved from social networks, applied to a mashup and web services composition. Before detailing our proposal, let's define clearly the notion of social network in our context:

Definition 1 (Social network): A social network is a graph representation of all the interactions that occur between people and services in a composition environment. This structure is automatically inferred from common interests between the users of the composition platform.

It comes from the definition that the social network we are considering at this stage is an implicit structure inferred from the common composition interests of users. Our framework is named *SoCo* (for Social Composer) and will be introduced in the rest of this paper. Figure 1 illustrates the general architecture of the framework. To help the user in the composition process, *SoCo* offers two main components: (i) a social knowledge extraction and modeling component and (ii) the recommendation manager. The first step consists of extracting and modeling the existing "knowledge" in a social network. Knowledge here stands for information which can be used in the context of services composition, e.g. which person uses which service in which composition. This knowledge is captured in order to define and construct a set of rules and actions that describe the composition behavior to improve and customize a composition in relation to a particular user. This is the role of the second component.

Additional components are also considered not necessarily visible on the figure: connection management with different social networks, a component for managing the users and the relations between them as well as the used (or created) services, and a service repository containing the list of the different available services. These additional components are useful since we want to exploit existing social networks. Therefore, it is necessary to handle the dialog between our system and the social networking platforms. The additional task will consist to relation generation between individuals since exploiting the existing relations directly is meaningless as we will see in Section III-A. In addition to a standalone web application, we envision framework as an external Web application embedded into the social networking platforms supported by, e.g., Facebook.

A. Social knowledge extraction and modeling component

Generally speaking, we consider two different ways to build social relations in this context: (i) explicit or (ii) implicit. For the explicit case, the user is offered the possibility to declare

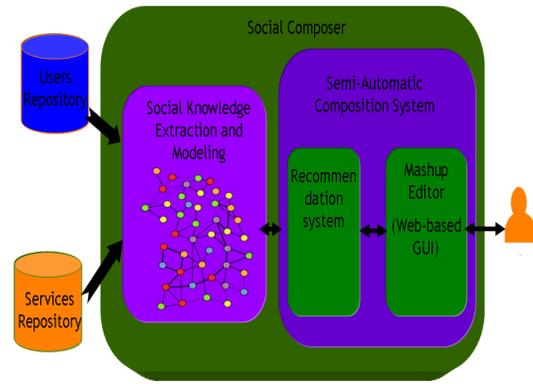


Fig. 1. General architecture of *SoCo* framework for social network integration in the composition process

a relation with another specific user. This is similar to what is done in today's social networks where two persons can become friends (i.e., socially connected) if the invited person accepts the invitation of the other. For the implicit way, a social relation is inferred according to the activities of the different users, e.g. when a person uses many of the created services of another person. In both the cases, we end-up with a graph linking the users according to their interests translated by their composition activities.

By construction, the information contained in a social network is specific and could be summarized in the following: (i) The user's profile that contains information that describes his special interests and preferences, and the history of his interactions with the system. Typically, these will include statistics on services utilization (consumption and composition). This information enables us to learn about the expertise of a given person in a particular area, and thus the relevance of services used by that person. (ii) The description of the links that define the social network itself. These links are used to calculate the social proximity between two people according to a particular context. This information allows us to calculate the service recommendation confidence between two individuals based on the specific joint interests. It is therefore very important to extract, analyze and model the information contained in a social network.

As outputs from extraction and analysis phase, we obtain two types of data: A profile of each user and the social proximities. The profile of a user will contain a number of metrics that describe interactions with services, in other words, his history of creating services and their uses. Among those information, we can find, for example, the number of services used in the composition, patterns of compositions of created composite service (sets of services succession). Social proximity is calculated on the basis of counting interactions between two users.

B. Service composition recommendation system

The main idea in this phase is to capitalize on information and knowledge from the previous phase to build a services recommendation system that is integrated into an overall

environment for services composition.

The recommendation system aims to help *SoCo* users during the edition of new mashups. It dynamically suggests services according to the current status of the services composition process, i.e. which service is likely to come after the current selected service? Thus, it intervenes in the services selection process. This selection considers different parameters related to the user's social network like the position of the user in his social network or more generally in the social graph, and information concerning the use of services by the social relatives and the social network.

More concretely, during the creation of a composed service through the *SoCo* service creation environment, a user generally is undecided about the selection of a service to follow a given service in the composition diagram. In this situation, the recommendation system will propose a list of services ranked on the basis of information provided from the social network analysis. Thus, the importance of a service recommendation is proportional to its use and also to the social proximity to the social relatives. This means that the more a service is used in this social network more the recommendation is important. Similarly, when users are close in the social network to the current user, the services they use are better recommendations (according to the current need). Moreover, when users who use certain services are experts, their choice is more relevant making the recommendation more important.

Consider For each user v_i is associated a local social network denoted by $SN(v_i)$ defined as (see Formula 1):

$$SN(v_i) = \{v_j \mid v_j R v_i, j : 1, \dots, n\} \quad (1)$$

The user v_i is creating a new composed service by editing a composition schema. For succeeding the current chosen service s_k , the system suggests a sorted list of recommended services presented in decreasing order of recommendation confidence RC .

We define RC , of a given service s_l for succeeding to the current service s_k , as follows (see Formula 2):

$$RC(s_k, s_l) = \sum_{j=1}^n NC_{v_j}(s_k, s_l) \times Fit(v_j, s_l) \times SP(v_i, v_j) \quad (2)$$

where:

- $NC_{v_j}(s_k, s_l)$ represents how many times user v_j has used s_l following s_k in composition diagrams;
- $Fit(v_j, s_l)$ quantifies the expertise of the user v_j in relation to service s_l , Fit is defined below;
- $SP(v_i, v_j)$ defines the v_i 's social proximity to v_j .

C. General implemented features

Regarding the different elements that need to be considered for implementing the *SoCo* mashups creation environment, we looked for an open source web GUI offering drag/drop functionality for mashups. We choose to use the *WireIt* Editor

Platform⁴ as a starting point for *SoCo*. *WireIt* is an open-source javascript library to create web wireable interfaces for dataflow applications, visual programming languages, graphical modeling, or graph editors.

Hereafter, we present the general features that our Framework provides or plan to, for both end-user and system dimensions: (i)From end-user viewpoint:The user interacts with the system through a GUI which enables him to express a services composition schema. The end-user is able to search, select, and attach services to create composite ones.

(ii)From system viewpoint: In addition to the core features, the *SoCo* framework will provide technology blocks to ensure the life cycle of a composite service, and interfaces with external APIs (such as REST architecture, SOAP / WSDL, etc.). Taken as a whole, the *SoCo* Framework is considering a composition and service model which defines the service inputs/outputs types, communications interfaces, etc.

IV. MOTIVATING USE-CASE

We discuss in this Section an illustrative use-case which exposes the different dimensions of the proposed framework. Consider Alice, a young fan of movies and cinema. To plan her cinema outings, Alice generally does many tasks before selecting the movie she is going to watch. For instance, she needs to consult many Web pages to get some information about the quality of a particular movie, ask her friends, locate a switable cinema to watch the movie. This is generally time and effort consuming. Alice is a novice in services composition even if she uses mashup editors but she has a good social community which can help her. However, her friends are generally not available to help her in building her service (mashup application). The *SoCo* system can be very of a great interest for Alice in this situation.

Thus, Alice decides to create a mashup that handles this kind of event using the *SoCo* to use composition habits of her social networks. Alice first searches for services related to cinema. She discovers that *FilmAdviser*, a service that offers movies on the basis of a set of preferences of users, is available in the services repository. She chooses the service and puts it on the *SoCo* editor window as shown in Figure 2. At that time, the recommendation system receives the query to find a possible service that can come after that service (i.e. *FilmAdviser*).Using the different information the system has about the social relations of Alice, the usage rate of the different services by that social network, etc. *SoCo* recommends *Cine-Map-Calendar*, a service which is generally used in the social network just after the selected service (Figure 2). *Cine-Calendar* is a service which considers a movie as an input and gives an output of the movie and its different schedules during the week. After that, *SoCo* proposes another service, *Cine-Map*, which, given the title of a movie and a specific city, draws the different cinemas where that movie is shown on a map. Alice decides to built the service composition and deploy it on mashup hosting environment (a browser).

⁴<http://javascript.neyric.com/wireit>

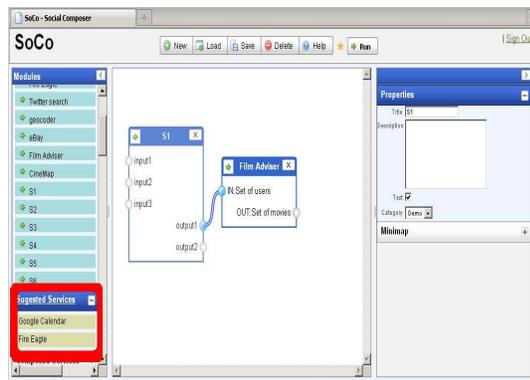


Fig. 2. A screenshot of services suggestion after selecting a second service SoCo

This use case presented above illustrate the potential value of the proposed framework in supporting end-users to create compositions of services, based on the analysis of social networks. These use cases also show the interaction of the framework with the end user.

V. CONCLUSION AND FUTURE WORK

We have discussed in this paper the necessity of considering social networks in the process of services composition and how this can help improve it. We have thus discussed different existing approaches for services composition and we have underlined the importance of semi-automatic composition which involves humans in the different levels of the composition process. We also showed how the emergence of Web 2.0 paradigms, namely user generated content, social networks, and collaborative work, is promoting this semi-automatic approach to services composition. This is particularly visible in the growing offering mashups creation environment where end-users can create their own services by aggregating other services and data sources. Regarding all those elements, we have introduced a new concept which uses informations issued from user's social networks in order to dynamically recommend him services composition schemes. Those recommendations take into account the current context of composition. This new concept involves the end-user's social network knowledge, the expertise of social network members and contextual social proximity between people (including trust). In that regard, we introduced a new framework, called SoCo, which aims at assisting the end-user in the process of creating new services by recommending her services based on the analysis of his social networks. We have detailed the components of the SoCo framework and illustrated it in a use case.

We believe that the longer the framework is used and the more it has information on how users compose services, the more the resulting recommendations will be relevant. Even if the analysis of this information is dynamic and allows the system to follow the latest trends in services selection and composition by social networks members, it is interesting to note the static aspect of social networks analysis which limits the system. By considering the skills of people and

their changing relationships in the social network, and that this information is directly involved in the calculation of recommendation confidence of a given service, it is necessary to adopt an updating strategy of the social graph which enables the system to calculate the recommendations on the basis of the most up-to-date information. Finally, an evaluation of the implemented system should be done either for users' viewpoint or system viewpoint.

REFERENCES

- [1] M. H. ter Beek, A. Bucchiarone, and S. Gnesi, "Web service composition approaches: From industrial standards to formal methods," in *ICIW*, 2007, pp. 15–20.
- [2] Y. Yuan, J. Wen, W. Li, and B. Zhang, "A Comparison of Three Programming Models for Telecom Service Composition," in *Proceedings of the Third Advanced International Conference on Telecommunications*. IEEE Computer Society Washington, DC, USA, 2007.
- [3] G. D. Lorenzo, H. Hacid, H. young Paik, and B. Benatallah, "Data integration in mashups," *SIGMOD Record*, vol. 38, no. 1, pp. 59–66, 2009.
- [4] J. Yu, B. Benatallah, F. Casati, and F. Daniel, "Understanding Mashup Development," *IEEE Internet Computing*, vol. 12, no. 5, pp. 44–52, 2008.
- [5] M. Broy, "Towards a formal foundation of the specification and description language sdl," *Formal Aspects of Computing*, vol. 3, pp. 21–57, 1991.
- [6] T. Law, "Social Scripting for the Web," *Computer*, vol. 40, no. 6, pp. 96–98, 2007.
- [7] L. Chen, N. Shadbolt, C. Goble, F. Tao, S. Cox, C. Puleston, and P. Smart, "Towards a knowledge-based approach to semantic service composition," *Lecture Notes in Computer Science*, pp. 319–334, 2003.
- [8] E. Sirin, J. Hendler, and B. Parsia, "Semi-automatic composition of web services using semantic descriptions," in *Web Services: Modeling, Architecture and Infrastructure* workshop in conjunction with *ICEIS2003*, 2002.
- [9] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara, "Semantic matching of web services capabilities," in *ISWC '02*. London, UK: Springer-Verlag, 2002, pp. 333–347.
- [10] S. Murugesan, "Understanding Web 2.0," *IT Professional*, vol. 9, no. 4, pp. 34–41, 2007.
- [11] D. Schall, H.-L. Truong, and S. Dustdar, "Unifying human and software services in web-scale collaborations," *Internet Computing, IEEE*, vol. 12, no. 3, pp. 62–68, May-June 2008.
- [12] L. Grammel and M. Storey, "An End User Perspective on Mashup Makers."
- [13] V. Hoyer and M. Fischer, "Market Overview of Enterprise Mashup Tools," in *Proceedings of the 6th International Conference on Service-Oriented Computing*. Springer, 2008, pp. 708–721.