Social Composer: A Social-Aware Mashup Creation Environment

[Extended Abstract]

Abderrahmane Maaradji, Hakim Hacid,
Johann Daigremont
Alcatel-Lucent Bell Labs France
Centre de Villarceaux Route de Villejust
91620, Nozay, France
{abderrahmane.maaradji, hakim.hacid,
johann.daigremont}@alcatel-lucent.com

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

ABSTRACT

Web2.0 emerging environment provides end-users with digital spaces, such social networks web sites, to share common interests and find lacking expertise offered by others. On the other hand we are witnessing an explosion in the number of reusable Web services and applications. Mashups creation environments enable end-users to reuse those Web applications to meet their needs. In this demo, we present Social Composer (*SoCo*), a CSCW system for mashup creation based on social-awareness approach. *SoCo* aims to support end-user when he is specifiying mashups to compose existing services in order to create a new ones through the mashup creation environment. *SoCo* provides services recommendation as autocompletion for user's partial mashup specification. Beyond existing systems based on tagging and rating, those recommendations are based on social networks implicitly built from the interactions between users and services, and the different services compositions operated by the user's social network members as well as the global social network.

Keywords

Social networks, Web Services Composition, Social Interactions Analysis, Social-Aware, Mashup Creation

1. TOWARDS A SOCIAL NETWORK BASED APPROACH FOR SERVICES COMPOSITION

Creating value-added services by reusing the existing service components, which is well known as service composition, has been a key issue service science and has been heavily investigated from both industrial and academic perspectives especially related to web services [4]. However, with the emergence of the Web 2.0, it is becoming more and more important to make the composition process much more end-user oriented. The semi-automatic services composition approach has the main advantage of making the user participate in the composition process. Indeed, Web 2.0 main paradigms are people-centric Web, participative Web, and read/write Web [2]. 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. Thus, 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. Semi-automatic Web services composition has taken several forms evolving over time from simple graphical tools to semantic-based tools. 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). 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*. Both community and social network oriented approaches aim at considering the produced knowledge to enhance the end-user service composition in cooperative environment. This generated knowledge could be either explicit as annotating, ranking, and rating services, or implicit knowledge by automatically processing extracted rules in order to build a recommendation system. Thus, several works have been launched around this area to exploit the knowledge of the mass in order to improve the composition process by considering either social networks or collaborative environments [3]. com A social network can not be considered in the community approach because it describes specific structures. The major difference is that 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 which can be available in social networks can be richer than the one of specific communities. On the other hand, Mashups are a new emerging paradigm of Web 2.0 with the aim of enabling end-users to easily create new web-based applications and services that address their specific needs and interests [5]. Several IT actors offer Mushups creation Web environment such as: Microsoft Popfly[1], Yahoo pipes[2], and Open Mashups Studio[3]. From the end-user point of view, Grammel and Storey [1] investigate tools and environments for creating Mashups. This analysis highlighted the need for a consideration of a community however it comes that all selected Mashup creation environment makers do not offer social networking features at all. To our knowledge, there is no a 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 perfectly with this issue. Our objective is to show how information can be extracted from social networks in order to be exploited in a composition task. in other words, *how to leverage social interactions in a way to enable and facilitate composite services creation?*. The next section discusses our ongoing attempt to contribute to this area by demonstrating our approach in Mashup creation environment.

2. SOCO: A FRAMEWORK FOR MASHUPS CREATION

SoCo is a CSCW mashup creation environment based on social-awareness approach that aims to support end-user when he is specifiying a composing schema using existing services in order to create a new mashup. *SoCo* provides services recommendation as autocompletion for user's partial mashup specification. Figure 1 illustrates the general architecture of our framework, called *SoCo* (for Social Composer). To help the user in the composition process, *SoCo* offers two main components: (i) social knowledge extraction and modeling component and (ii) the recommendation manager.

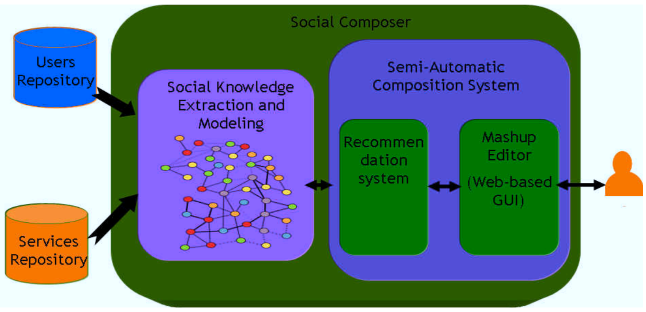
Social Composer
Users Repository
Services Repository
Social Knowledge Extraction and Modeling
Semi-Automatic Composition System
Recommen dation system
Mashup Editor (Web-based GUI)

Figure 1: General architecture of the social networks integration in the composition process (*SoCo*)

Social networks we are considering at this stage is an implicit structure inferred from the common composition interests of users. At outputs from extraction and analysis phases, 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.

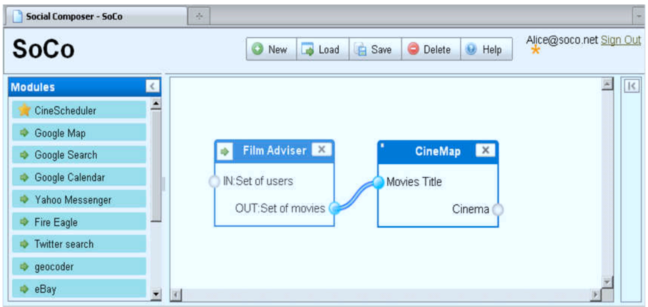

Figure 2: A screenshot of services suggestion after selecting a second service *SoCo*

The recommendation system aims to help the *SoCo* user during the creation of a service by proposing services to compose according to the current status of the service composition process (i.e. which service can be better to come after the current selected service).

More concretely, when it comes to the creation of a service, i.e. composed service, through the *SoCo* service creation environment, a user generally is undecided about the selection of a service as a successor for 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 as shown in Figure 2. The ranking process is made according to the recommendation rank of each service. Thus, the importance of a service recommendation is proportional to its use on one hand and to the social proximity to the social relatives on the other hand. This means that more a service is used in this social network more the recommendation is important. Similarly, more users are close in the social network to the current user, the services they use are better recommended (according to the current need). Moreover, more users who use certain services are experts, their choice is more relevant making the recommendation more important. A more detailed use case in described in the demo system summary.

3. REFERENCES

[1] L. Grammel and M. Storey. An End User Perspective on Mashup Makers.

[2] S. Murugesan. Understanding Web 2.0. *IT Professional*, 9(4):34–41, 2007.

[3] D. Schall, H.-L. Truong, and S. Dustdar. Unifying human and software services in web-scale collaborations. *Internet Computing, IEEE*, 12(3):62–68, May-June 2008.

[4] M. H. ter Beek, A. Bucchiarone, and S. Gnesi. Web service composition approaches: From industrial standards to formal methods. In *ICIW*, pages 15–20, 2007.

[5] J. Yu, B. Benatallah, F. Casati, and F. Daniel. Understanding Mashup Development. *IEEE Internet Computing*, 12(5):44–52, 2008.

[1] http://www.popfly.com/

[2] http://pipes.yahoo.com/

[3] http://www.openmashups.org/

Social Composer: A Social-Aware Mashup Creation Environment

[System Summary For Reception Demo]

Abderrahmane Maaradji, Hakim Hacid,
Johann Daigremont
Alcatel-Lucent Bell Labs France
Centre de Villarceaux Route de Villejust
91620, Nozay, France
{abderrahmane.maaradji, hakim.hacid,
johann.daigremont}@alcatel-lucent.com

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

ABSTRACT

Web2.0 emerging environment provides end-users with digital spaces, such social networks web sites, to share common interests and find lacking expertise offered by others. On the other hand we are witnessing an explosion in the number of reusable Web services and applications. Mashups creation environments enable end-users to reuse those Web applications to meet their needs. In this demo, we present Social Composer (*SoCo*), a CSCW system for mashup creation based on social-awareness approach. *SoCo* aims to support end-user when he is specifiying mashups to compose existing services in order to create a new ones through the mashup creation environment. *SoCo* provides services recommendation as autocompletion for user's partial mashup specification. Beyond existing systems based on tagging and rating, those recommendations are based on social networks implicitly built from the interactions between users and services, and the different services compositions operated by the user's social network members as well as the global social network.

1. THE SOCO FRAMEWORK

The SoCo framework (for Social Composer) is a CSCW mashup creation environment which provides facilities to end-users at the service selection level by providing recommendations based on processed knowledge that has been extracted from users' social network. Figure 1 shows the functional view architecture of our framework. To help the user in the composition process, *SoCo* offers two main components: (i) social knowledge extraction and modeling component and (ii) the recommendation manager.

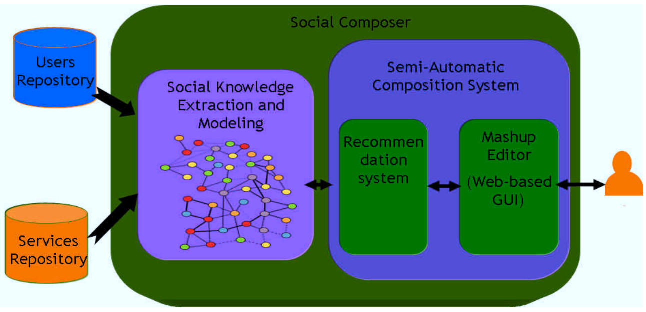

Figure 1: General architecture of the social networks integration in the composition process (*SoCo*)

At outputs from extraction and analysis phases, 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.

The recommendation system aims to help the *SoCo* user during the creation of a service by proposing services to compose according to the current status of the service composition process (i.e. which service can be better to come after the current selected service).

More concretely, when it comes to the creation of a service, i.e. composed service, through the *SoCo* service creation environment, a user generally is undecided about the selection of a service as a successor for 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. The ranking process is made according to the recommendation rank of each service. Thus, the importance of a service recommendation is proportional to its use on one hand and to the social proximity to the social relatives on the other hand. This means that more a service is used in this social network more the recommendation is important. Similarly,

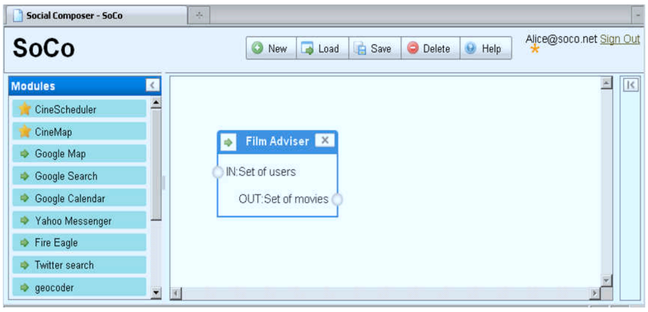

Figure 2: A *SoCo* screenshot of services suggestion after selecting *FilmAdviser* service

more users are close in the social network to the current user, the services they use are better recommended (according to the current need). Moreover, more users who use certain services are experts, their choice is more relevant making the recommendation more important.
In addition to services recommendation as the main feature, we plan to integrate common features in order to fulfill the *SoCo* framework. Planned features are: (i) capabilities for tagging, rating, ranking services, (ii) a learning support, and (iii) abstraction levels for service presentation to meet end-user skill levels (average web-user, advanced or programmers). Another innovative feature, we are planning to integrate, is services browsing based on users' social network which will allow users to adopt social network-driven services composition approach.

1.1 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 Platform [1] as a starting point for *SoCo*. textitWireit is an open-source javascript library to create web wirable interfaces for dataflow applications, visual programming languages, graphical modeling, or graph editors. WireIt is tested on all A-Grade Browsers, although it might work with older versions of browsers and platforms. It uses the YUI library (2.7.0) for DOM and events manipulation, and excanvas for IE support of the canvas tag. The code for Wireit is provided under a MIT license.

The SoCo requires a connection to a database to use save users' composition habits. It uses JSON-RPC through ajax calls, and a PHP/MySQL backend hosted on EasyPHP web server. The user interacts with the system through a bowser-based GUI which enables him to express a services composition schema. The end-user is able to search, select, and attach services to create composite one.

2. MOTIVATING USE-CASE

We discuss in this Section an illustrative use-case which aims at exposing 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 good cinema where to watch the movie. This is generally time and effort consuming making Alice watching any movie in the cinema. Alice is 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.

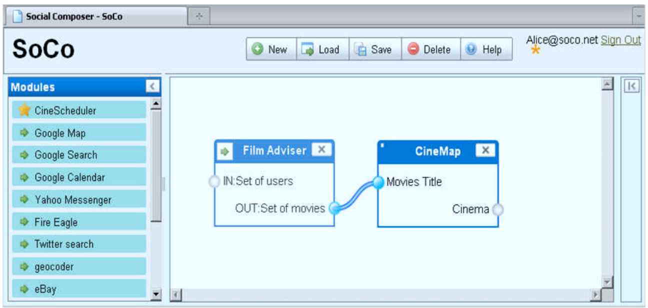

Figure 3: A *SoCo* screenshot of services suggestion after selecting a second service, *Cine-Map-Calendar*

Thus, Alice decided to create a Mashup that handles this kind of event using *SoCo* to benefit from composition habits of her social networks. Alice first searches using a query of services related to cinema. She discovers that *FilmAdviser*, a service which offers movies on the basis of a set of preferences of users is available in the services repository. She then chooses the service and put 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 which can come after that service (i.e. *FilmAdviser*).requests are automatically sent to the recommendation system to suggest a service that could precede the service *FilmAdviser*.

Using the different information the system has about, e.g., 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 3). *Cine-Calendar* is a service which considers a movie as an input and gives an output of the movie and its different planning during the Week. After that, *SoCo* proposes another service, *Cine-Map*, which, given the title of a movie and a specific city, returns the different cinemas where that movie is shown on that city. Next, the system also recommends the use of a service to buy cinema tickets . Alice prefers to buy the tickets when she gets to the cinema and thus she don't use that service and decides to built the deploy it on Mashup hosting environment (a browser).

3. THE DEMO PRESENTER

Abderrahmane Maaradji is a PhD student at Bell Labs with collaboration with Institut Telecom. His research interests cover web services composition, user services creation, social networking, social interactions analysis, Mashup Creation. In the Social Communications department, he is a member of the team in charge of the SoCo framework development.

[1] http://javascript.neyric.com/wireit