SOCIAL NETWORK-BASED FRAMEWORK FOR USERS AND WEB SERVICES DISCOVERY

Hiba H. Fallatah

A Thesis

in

The Department

of

Concordia Institute of Information Systems Engineering

Presented in Partial Fulfilment of the Requirements for the Degree of Master of Applied Science(Information System Security) at

Concordia University

Montréal, Québec, Canada

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Hiba Fallatah

Entitled: Social network-Based Framework for Users and Web Services Discovery

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science(Information Systems Security)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. R. Glitho ______ Chair

Dr. S. Klasa ______ Examiner

Dr. A. Ben Hamza ______ Examiner

Dr. J. Bentahar ______ Supervisor

Approved by ______
Chair of Department or Graduate Program Director

______
Dean of Faculty

Date March 13, 2014

ABSTRACT

Social Network-Based Framework for Users and Web Services Discovery

Hiba H. Fallatah

With the emergence of Web 2.0 and its applications, social networks have facilitated the discovery process of web services, a cornerstone to the development of service computing. Very recently, some frameworks have suggested adding a social element to services' description, discovery, binding and composition. By incorporating the social component in the Service-Oriented Architecture, web services become active entities that can form and be part of social networks. However, merging users and web services in the same social network and analyzing the influence of these entities (i.e., web services and users) on each other have not been examined in the previous proposals and yet to be investigated.

In this thesis, we propose a new social network-based framework for analyzing the role and influence of users and web services in the discovery process. We advocate the idea of incorporating, not only social web services, but also social users in the discovery process by merging users and web services nodes in the same global social network. We first discuss the engineering process of such a social network that takes into consideration users and web services characteristics and the types of their interactions. Thereafter, we analyze those types of interactions that fall in one of two categories: web service discovery or user discovery. The goal is to involve social networks of users in the service discovery process and allow web services to be active parts by advertising and introducing themselves to other users. Simulation results show that the proposed approach provides

an immediate and wider exposure for web services and makes the discovery easier and efficient.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting the research and the writing of this thesis. First, my supervisor, Dr. Jamal Bentahar, for his guidance, patience and support throughout the writing. I would also like to thank my lab mates for their help and support. Additionally, I thank Mr. Ehsan Khosrowshahi Asl for his precious help in the implementation process. Greatest appreciation goes to my parents for their love and support all through my studies. Finally, I would like to express my sincere gratitude to my brother Haitham, my husband Wajdi and my son Yazan for their patient and continuous encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

With the growing use of Services-Oriented Computing (SOC) applications, namely web services, researchers have focused on the importance of the discovery problem. Service discovery is the automatic detection of services on the web by end users or by other services. Current discovery methods are registry-based methods that rely on syntactic or semantic description of web services [1, 21, 33, 40]. These methods suffer from low precision results mainly because web services are handled as isolated elements [27], and there is no consideration of the user role or needs in the service requesting process. Recently, some research groups launched the idea of incorporating social aspects in service computing [25, 27, 28, 29, 48]. Using social computing to build *social web services* has helped the discovery process overcome some of these issues [26, 27]. However, the potential of social networks has not been fully exploited yet since the main component, the user, is not yet part of the solution. Thus, investigating the effect of extending the social concept to include users to be interactive entities in this social settings is still to be addressed, particularly from the efficiency perspective.

In this chapter, we will introduce the context of the research, which is about using social network metaphor as a tool of discovery. We also refer to our motivation behind conducting this research, which is to make use of the diverse information captured in the social networks to improve the discovery process, in addition to boosting the web services' advertising capability. The two later sections of this chapter are about our contributions and thesis organization.

1.1 Context of the Research

Web services are active software programs that are mainly designed to support machine to machine interactions through the web. Web service technology is one of the main implementations of the SOA that has changed the way software has been designed and developed in today's modern, across enterprise-oriented applications [26]. As services providers need to advertise for their web services, they publish them in the discovery registry model UDDI (Universal Description Discovery and Integration) to facilitate the process for the user to find the required service. However, the use of UDDI has faced many difficulties because of its centralized structure and its use of static description for the services, which makes searching, detecting, and localizing web services a big challenge [39]. As the number of web services grew rapidly, the need for an effective method of services discovery for user needs and service advertising have become more essential. Social web services have been introduced to overcome these problems [28, 27, 30, 29]. These networks have been formed according to the services functional similarity or complementarity. Having social web services in an enterprise setting can help those services exhibit human-like social behaviors, in which they could collaborate or substitute depending on their needs and functionality. Moreover, adding the social element can benefit users by helping them become proactive entities that can request services, as well as provide tips about and evaluation of the services they have used [26].

This thesis focuses on enhancing the discovery process of web services using both users' and web services' social networks. One of the challenges is to capture the behavior and various interactions between social network peers, as well as the impact of merging different nodes in the same global network. As web services social network has assisted the process of finding new peers [27], users social network influence in this setting as a discovery tool and its integration with web services network are still unclear and will be addressed in this thesis.

1.2 Motivation

The motivation of this thesis is to capture the knowledge generated through users' and web services' interactions in their respective social networks to enhance the web services discovery process, make services selection more end user oriented, and expand the exposure of web services among other users and other peers. Usually, web services are found by screening discovery directory (i.e., UDDI). As the number of web services is continually increasing, using UDDI does not allow us to take into account web services' history of interactions and experiences. This information is very useful in capturing information that could help find higher quality services for users. One way to capture this information is by using the structure and metaphor of social networks. A social network has the ability to authorize web services to recommend the peers that they would prefer work with in a composition, or recommend the peers that they would like to substitute them in case of failure [29]. To reach our goal, we propose in this thesis a new framework that enables users' and web services' social networks to act as a method of discovery for web services.

1.3 Contributions

This thesis has two main contributions:

- The first contribution is developing a framework that would enable web services to discover new peers and add them to their social network with the aid of the users' social network. This technique will assist web services expand their social connections, with respect to their functionalities, using their connections to users' nodes. This will results in more personalized services for the end user who become more involved in the service selection process. This framework can also serve as a discovery method for users' nodes as well, in which users can find their peers by screening their connections to web services.

- The second contribution aims to provide an analysis of the users and web services behaviors inside their social networks and how this behavior could affect the ability to find the required services. This analysis can give us an insight on how growth of the social network could be affected. Furthermore, such an analysis will contribute in enhancing the advertising process by allowing web services to discover and introduce themselves to new users through other users and web services.

1.4 Thesis Organization

The thesis is organized as follows: This chapter discusses the context of the research, motivation and contributions that this research will add to the state of the art. Chapter 2 presents the background, which includes some introductory information about web services, social networks, social computing, service oriented computing, and finally a literature review of previous related work. Chapter 3 introduces and discusses our model of global social network for the discovery process. First, we demonstrate the relationships of the social users and web services within the context of social networks engineering. Thereafter, we list four scenarios that could come up in using social networks as our tool of discovery. Chapter 4 discusses the simulation process and experimental results. The conclusion of this thesis and future work are presented in Chapter 5.

CHAPTER 2. BACKGROUND AND LITERATURE REVIEW

In this chapter, we present and review general information that will help us understand the framework presented in later chapters. The organization of this chapter is as follows. Section 2.1 introduces our work background. It presents web services definitions and concepts and discusses social network and its value and characteristics. Social computing and social oriented computing are also highlighted in the same section. Section 2.2 presents relevant related work in the field of web services discovery. Engineering social networks and their role and value are discussed in the same section as well.

2.1 Background

2.1.1 Web Services

A- Definition and Main Characteristics

Web service is defined in The World Wide Web Consortium (W3C) as a "software system with interface described in Web Services Description Language (WSDL)". This software is designed and built to support machine-to-machine communication through the network. Simple Object Access Protocol (SOAP) messages are used to enable web service to interact with other systems. These massages are transported through HTTP (Hyper Text Transfer Protocol) with XML (Extensible Markup Language), and associated with other web standards [5]. XML is used to represent web resources [7]. It "allows web service operation to be accessible through the network" [13]. XML is used to describe WSDL to illustrate and describe the service required. This description includes a message and its formats, its location and the protocol used to transport this message [10]. In order to discover these services, the description can be published in UDDI [44] through services registry. SOAP is mainly used as communication protocol for interaction between applications and web services to exchange structured and typed information [6]. In order to convey SOPA messages, HTTP is usually used as the transporting protocol but other transport protocols can be used as well. The interface should be designed in a way that exposes all the needed information to use the service appropriately, and hide unnecessary details.

B- Web Service Architecture

Figure 2.1 illustrates the web service architecture. In this architecture, service provider will host a web service software that can be accessed through the network. Also, the provider describes the services and publishes the description in the UDDI registry based on its specification. After publishing the web service in the UDDI, the service requester can find the web service through the registry interface. The service requester will obtain service description in WSDL form, in addition to the web service URL from UDDI. This information will be used by the services requester to use the web service of his choice.

2.1.2 Social Network

A- Definition

When we refer to social network in the most basic sense, we focus more on the social structure that consists of “actors” and “ties” (or nodes and edges) between them. The actors are the individuals or organizations in the social network, and the ties represent the connections and relationships between these actors [47]. In another definition, a network is any collection of objects in which some of these objects will be connected by

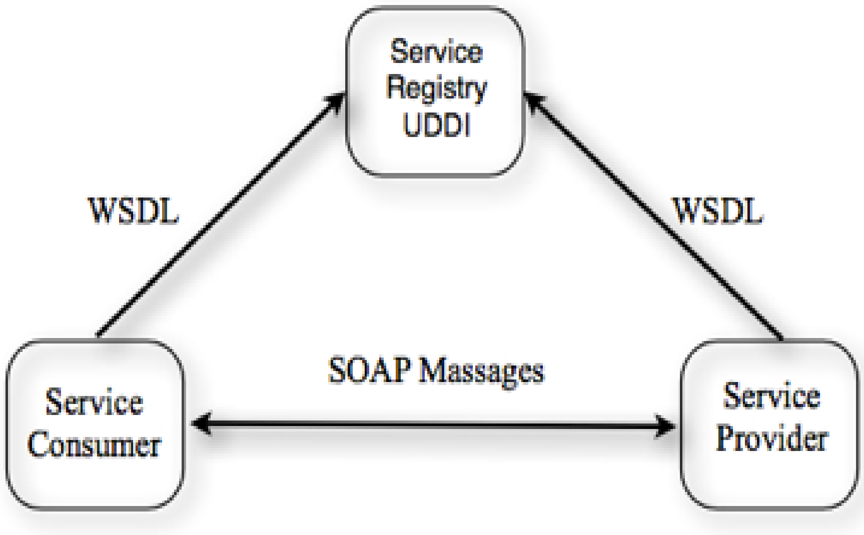

Figure 2.1 Web service architecture

links [12]. Both definitions are wide and flexible, which can be easily applied in many domains.

To have a simple representation of what a social network should look like, we examine Figure 2.2, which portrays the social network among 34 individuals in a university karate club. In this network, students are represented by small circles (nodes), with lines for the edges joining the students who are friends outside the club context [12]. This social network has been staged by the anthropologist Wayne Zachary in the 1970s [49].

It is worth mentioning that the terms *network* and *graph* are used equivalently in the scientific literature, although they are quite different terminologies. Network is a collection of nodes and link combinations that represent a real or virtual system. A typical example is the WWW, which is a network of web pages connected by URLs. Another example from society is a network of individuals connected by family, friendship or professional ties. On the other hand, graphs, composed of vertices and edges, are used to refer to mathematical representations of these networks such as the web graph and

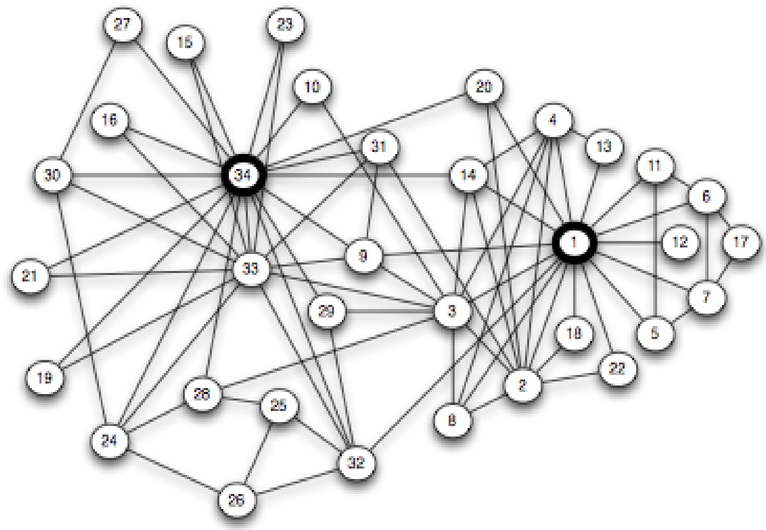

Figure 2.2 Social network of friendship within a 34-person karate club [49].

the social graph. Still this distinction is rarely made, though both terminologies are used as synonyms of each other [3].

B- Value of Social Users

In the real life, people form circles and communities based on their relationships, expertise, experiences, age and gender. These communities or networks can be transferred to the digital world. With Web 2.0 becoming the new face of the Internet, the structure of communication between the social networking entities and the Web has changed the way users (or entities) can play various roles simultaneously. Users can not only read or download passively, but also write or upload content actively [16]. In a business setting, users can play the role of consumers and providers at the same time [16, 35]. A major part of social networking is sharing opinions, expertise, and interests, as well as collaboration, skills-building and discovery [16, 37]. In order to share, collaborate or even discover, a recommendation method should be used. Recommendation systems can be classified either as content-based or collaborative filtering [9].

- In the content-based class, the user will have a recommendations for items similar to what he had picked or preferred in the past. This method depends on item descriptions that could be created using machine learning or information retrieval techniques.

- On the other hand, collaborative filtering depends on a collective of user preferences of items. In this method, a set of users provide the preference information to be used to make automatic personalized predictions for the active user.

C- Social Network Characteristics

Although social networking services can be developed for different purposes, they also have features or characteristics in common [2, 8]:

- Massive connections: many users are able to be connected and attract new connections.

- Multi model interactions: to support a lot of interaction methods through the web such as voice chatting and instant massaging.

- Self organizing collaboration: users cooperate and work together for the same target or common objectives.

- Self-motivated participation: to motivate users to enrol in a focus group or community that matches their interests and preferences and then make a contribution to benefit other users.

- Sharing of user-generated content: to stimulate creativity by motivating users to share materials made by them.

2.1.3 Social Computing

A- Definition

Enabling the online interactions and exchanging information and multimedia through the Internet is what we refer to as social computing [37] . In [45], social computing is related to the study of the human social dynamics, as well as information and communication theologies design and use. Social computing is also a term used in computer science to refer to the interactions among software entities that involve any kind of social behavior.

B- Infrastructure of Social Computing

Computational and social sciences are the core of the social computing infrastructure, and so it can be looked at from a variety of perspectives (See Figure 2.3). It could include web database, multimedia, wireless, agent, and software engineering technologies as an information processing prospective. On the other hand, the methodological perspective merges social theories with technology development and usually leads us to use additional requirements to construct artificial societies by using agent modelling techniques [45].

C- Major Application Areas

The aims of social computing applications is to develop a better social software that makes interactions between social entities easier, computerize some aspects of human society, and also to forecast the effect of changing technology and polices on social and cultural behaviors [45]. Social computing applications can be categorized into four areas:

- Web 2.0: witch supports effective online communications for social communities.
- Entertainment software: this application focuses on building intelligent entities that can interact with human users.

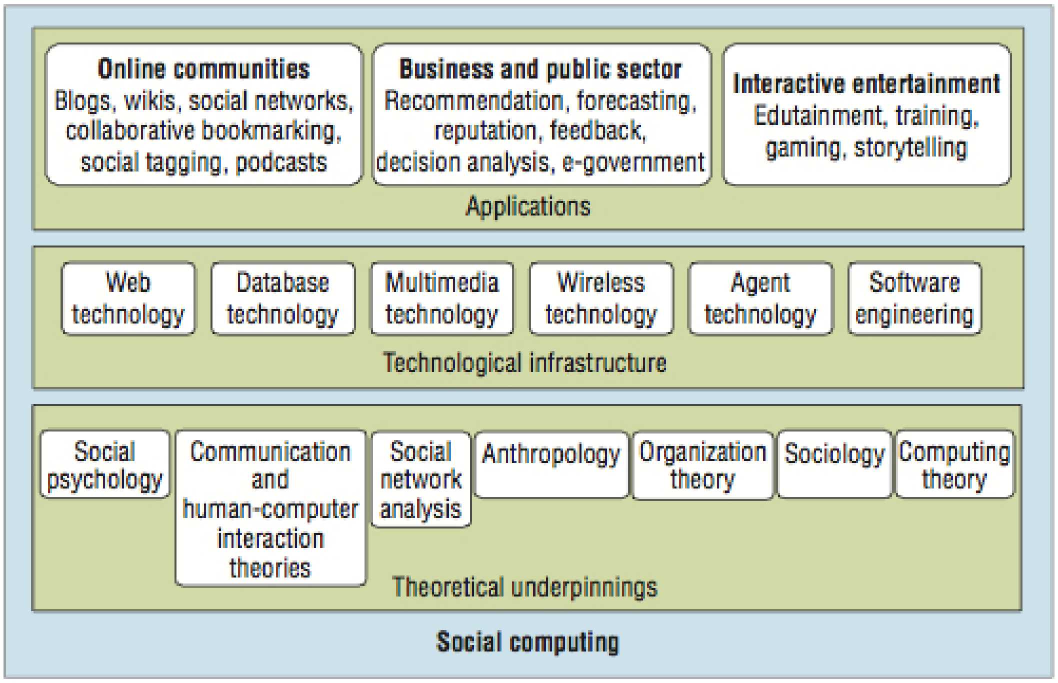

Figure 2.3 Social computing infrastructure and applications [45]

These two applications value the technological part more and use the social theories part as guidance for both designing and framing computing systems.

- Business and public sector: such as E-business, healthcare, economics, political and digital government.
- Forecasting: this application includes predictive systems for planning, evaluation and training [45].

2.1.4 Service-Oriented Computing

A- Definition

SOC is the computing design paradigm that uses services as fundamental elements for developing applications [4]. SOC builds connections between technology and enterprise organizations [4], which means, it will help design software services that will be used to advertise for business.

B- Services-Oriented Architecture and Requirements

Recently, the implementation technology using Services-Oriented Architecture (SOA) has become the focus of enterprise applications and development. Using SOA for designing a software system is a logical method to provide services to the end-user's application or other services using discoverable interfaces [36]. SOA in its basic form defines an interaction between software agents, which are the services requester (client) and the services provider, as they exchange messages. The different elements of service-oriented architecture are shown in Figure 2.4

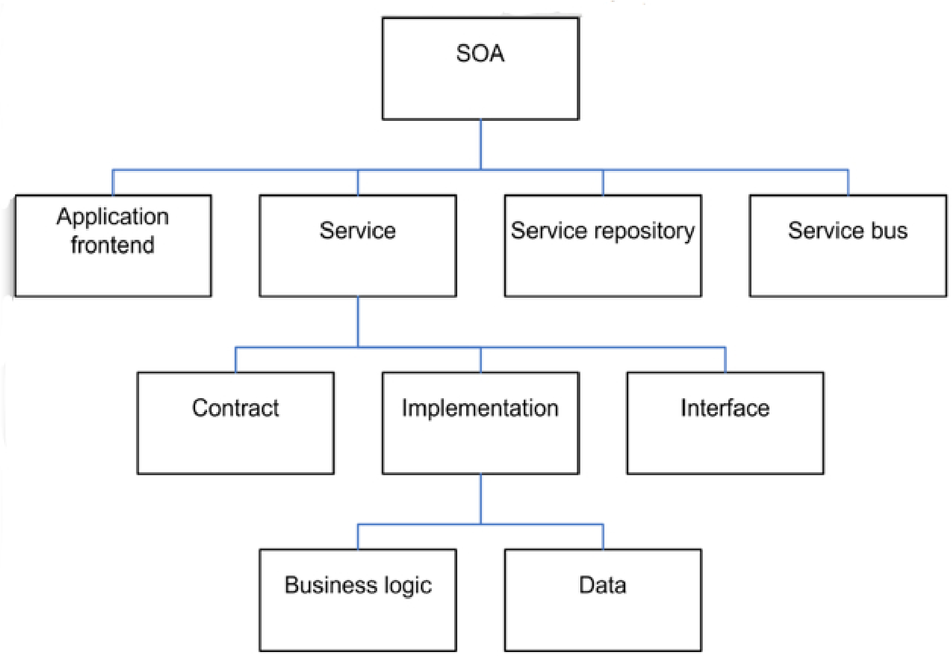

Figure 2.4 Elements of Service-Oriented Architecture [20]

The SOA demands the following requirements [41]:

- Loose coupling: there should not be tight transactional properties that can be applied among the components.

- Implementation neutrality: the interface is the most important part here not the implementation. Implementation details of the interacting component are not the right choice to depend on.

- Flexible configurability: system components are bound to each other late in the process, so that the configuration can change dynamically.

- Long life time: the component must exist long enough to be able to eliminate any relevant exceptions, to take corrective action, and to be able to respond to the corrective actions taken by others. A component must exist long enough to be discovered, relied upon, and to generate trust in its behavior.

- Granularity: capturing the essential high level qualities that are visible, is a better approach than modelling actions and interactions at a detailed level.

- Team: computations in open systems is more about business partners working as a team rather than about commands being issued and obeyed. That means instead of an individual, a team of cooperating participants is a better model.

By combining social computing with service oriented computing, we can build applications that allow users to collaborate with one another and share data. Besides, users will be offered a large menu of services from which they can request a certain service from providers. All this will form social web services that will be able to interact by collaborating, substituting and negotiating between their peers and with users. In the next chapter, we will present our social network-based framework for both social users and web services and discuss details about their interaction types and how web service discovery will benefit from that framework.

2.2 Literature Review

2.2.1 Web Services Discovery

The purpose of web services discovery is to enable the requester to find a web service that meets the need [43]. In [27] and [43], the authors refers to three approaches to discovering web services:

1. *Structural discovery.* This is a technical matching that requires the service requester (user) to specify the structural requirements such as data type. This approach uses the available syntactical information, as the definition of the data messages traded between partners, or uses term-frequency or inverse-document frequency to spot the most similar web service to the user's description as suggested in [1, 40]. These approaches do not expose the semantic relationships between web services.

2. *Lexical discovery.* This approach refers to the natural language description for the terms used to describe web services operation names' functionality. Usually, what the lexical algorithms do is remove stop words from the descriptions, then find synonyms through lexical database such as WordNet [33] and then compute the similarity coefficients. The Universal Description, Discovery and Integration (UDDI) is an example of this technique, where it uses a keyword-based search, although this process is time consuming and ineffective.

3. *Semantic discovery.* Semantic description often based on ontology. As in [21], it describes web service capabilities and properties using formal methods, so the possible candidate for the requester can be identified using machine reasoning.

Also, in [43], the authors list other methods to classify web services discovery approaches: early binding, and late binding. They differentiate between discovery time during design and run-time. Early binding is initiated during web service composition design by the user. In late binding, which is used during services composition execution, discovery of all the matching web services will be done at the run-time. Composition does not state which web services will be used, but it has services call definition requirements.

Unfortunately, all the discovery approaches referred to deal with web services as isolated entities that have no interactions between them, although this kind of interaction is required to build and complete compositions [26, 27].

In [29], the authors have develop a model called LinkedWS to discover web services using social network concepts. This model allows web services to be engaged in two different types of social network: a) a collaboration social network when web service takes part in composition to provide the user with a complex request; b) recommendation social network, which is split into robustness and partnership social networks. Partnership enables a web service that is already joined in a composition to recommend the user to add additional peers that will grant more responses in the composition. Robustness, allows the web services to identify the peers that can substitute for it in case the web services fail or are not able to respond to the user request. Web service can know these peers based on their functional and non-functional similarity.

However, this approach has not illustrated the role of the user in the discovery process, although the user is the one who we want to provide the service to [30].

2.2.2 Engineering Social Web Services

Many researchers have analyzed the web services social network, its relationships and behavior under different conditions. Some of their papers give details of constructing social web services and analyzing their behavior [25, 28]. The authors of [28] discuss the structure and management of a web service community by focusing on intra-community coopetition to sustain community growth and inter-community competition to make similar communities compete to attract more users and providers. The authors blend both social computing and SOC to construct social web services. They hypothesize an engineering method in two phases:

- First Phase: Analysis, which includes two steps:

1. Establish the relationship between web services. They have discovered three types of relationships:

   (a) *Competition.* When web services compete with each other to be selected.

   (b) *Substitution.* In case of failure, web services substitute each other in order to continue serving the user.

   (c) *Collaboration.* Web services collaborate in order to deliver a composite service to the user.

2. Map the previously established relationships into the social network. Three types of social networks according to the identified relationships are distinguished:

   (a) *Competition social network:* To make web services aware of their competitors.

   (b) *Substitution social network:* To make web services highly available in case of failure.

   (c) *Collaboration social network:* To keep track of all the web services that worked in the compositions.

- Second Phase: Design, which has the following two steps:

1. Define the characteristics of each social network (i.e., number of nodes, type of edges, and weight matrices for these edges). Social networks have two components: nodes(corresponding to the web services), and edges(corresponding to relationships between these nodes) that will form the three previous mentioned types of social network. Here, we will only focus on substitution and collaboration social networks.

Substitution social network. This social network has web services that match in their functionality. However web services in the substation *SN* replace each other in case of failure. Some of the edges in this *SN* are bidirectional edges, and the wight of these edges can be computed using *(Substitution Level)* $SubL_{ws_i,ws_j}$.

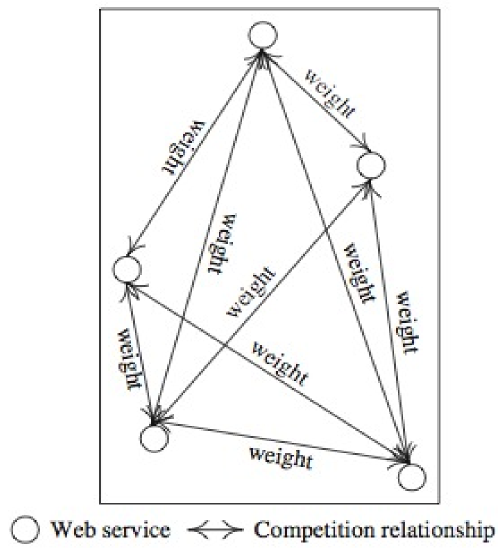

Figure 2.5 Competition social network [28]

$$SubL_{ws_i,ws_j} = FSL_{ws_i,ws_j} \times RL_{ws_i,ws_j} \times (1 - NFSL_{ws_i,ws_j}) \qquad (2.1)$$

where FSL_{ws_i,ws_j} is the Functionality Similarity Level to measure the functional property similarity between two web services, ws_i and ws_j [11]. $NFSL_{ws_i,ws_j}$ is the No-Functionality Similarity Level to compare web services non-functional property [28]. And $RL_{ws_i,ws_j}$is the *Reliability Level* that shows how successful ws_i in replacing ws_j[28].

Collaboration social network. In this social network web services have different functionality. This social network SN is built when one web services composition at least is complete. All the edges here are unidirectional edges, and come from *focus web service* and pointing to the other service

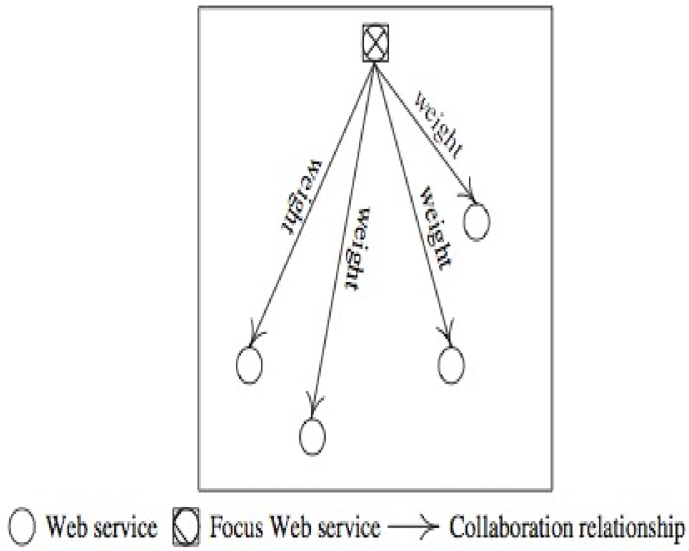

Figure 2.6 Collaboration social network [28]

($ColL_{ws_i,ws_j}$) *Collaboration Level* is used to evaluate the weight of the collaboration edge.

$$ColL_{ws_i,ws_j} = \frac{\sum JC_{ws_i,ws_j}}{\sum TP_{ws_i}} \tag{2.2}$$

where $\sum JC_{ws_i,ws_j}$ is the total number of the joint competition that the two web services ws_i and ws_jhave participated in. $\sum TP_{ws_i}$ is the total number of participation of ws_i in compositions.

2. Analyze the social behavior that a web services exhibits in being part of these networks. There are many types of social behavior that the web service can exhibit depending on its social network.

Selfish social behavior. This type of behavior is found in the substitution social network, where a certain web service ws_j refuses to replace failing peers (i.e. ws_i). However, ws_i is continually replacing ws_j when its fails.

Malicious social behavior. Web services that exist in competing social network can exhibit a malicious behavior. Usually, when a web services refuses to handle user requests, it sends that user to another web service to handle that user. If that web service, when accepting to serve the user, is not sure about its Quality of Service (QoS) level [11] and still accepts the user request, it will be considered a malicious web service.

Dominant social behavior. In a collaboration social network, a *ws* is considered dominant over another peer if the *ws* engages in a composition with that peer more than the peer does with the web service.

2.2.3 Role of Web Services Social Network

A- Web Services Drawbacks

In the previous section, we mentioned that the discovery approaches used deal with web services as isolated components that have no interaction between them, or do not take users' knowledge into consideration. That leads us to the conclusion that the discovery process should take into account not only the desired web service, but it should consider its user and peers interactions as well. To illustrate the importance of this point, Z. Maamar et al. list in [26] web service characteristics when they operate as separate entities:

- web services do not know anything about their peers or users, they only know about themselves.
- because they function as a black box , users' involvement is restricted.

- during execution, web services take in to account only their own functional and non-functional details, and disregard such external details as past user interactions.
- they can not delegate their invocations
- they can not cooperate or organize themselves.

As a result, the SOA situation has limited the use of web services because of many unresolved issues [26] such as:

- how to find the best location for immediate exposure to advertise for their services
- considering user needs in service discovery.
- trusting the discovered services.
- replacing services in case of failure.

B- Value of Social Web Services

Social network is a typical example of Web 2.0 applications, in which it helps users to become actively involved in offering and requesting services at the same time. Using social networks in different fields (e.g, artificial intelligence, business) has advantaged individuals and groups in today's society. In distributed artificial intelligent forming coordination, corporation and negotiation is possible using social network [29]. Also, social networks topologies and an agent's location in his social network can determine the agent's reputation [38]. Social network used in recommendation systems in [32]provide recommendations to users looking for collaboration appropriate to the offered and needed expertise. In addition social network can help solve specific problems by finding the best expert individual who would be able to find the right solution[50].

With all that being said, adding the social element to web services description, discovery, binding and composition will support SOA needs, and these new social web services *(SWS)* will [26]:

- be a network of contacts that can be established and maintained
- have additional functionalities for web services that can be enabled through collaboration
- depend on privileged contacts when required
- build long-lasting solid collaborative groups with other peers

On the other hand, users should be considered in this web services composition, in other words, composition should be more end-user oriented [30]. This consideration has made web 2.0 offer different technologies for users who have limited expertise in programming to facilitate their use of web services and its applications. In [30] the authors identified three ways that web services composition can be performed: (1) Manually where services are entirely produced by hand; but this method required a high programming technical level from the user, which makes it of limited value; (2) Automatically where the goal is to build automatically composite services that match the user context or request, but there is no involvement of the user during the composition creation; or (3) Semi-automatically, which offers the user a support environment in order to process some of the composition tasks in an automated way, and the user can operate with less or more involvement.

Consequently, the semi-automatic approach has been successful in making the user participate in the composition process and at the same time keeping his overhead very small, and that can happen by making full use of the information generated by the user or his community. The authors in [30] have also categorized this approach in three categories:

- **Single end user-oriented** This approach builds a user profile that contains information about the user in the non-decision stages, and tools/and interface to help the composition process of the applications.

- **Domain or community oriented** His takes in to account the knowledge created in communities or in a specific domain, and it can be either (1) in the pre composition process (at discovery and selection levels), or in the post composition process (when the service annotated ranked and rated); or (2) by extracting the existing knowledge of this community or domain using a process that defines a set of rules and builds a recommendation system for composition.

- **Social network oriented** The reason that this approach is not included in the community approach is that the knowledge available in a social networks can be richer than that available in a particular community. In a sense, social networks are friends' networks, and are constructed on the basis of specific interest for each relationship in the network, but the community knowledge is gathered from individual's who have interest in a common topic.

CHAPTER 3. USERS AND WEB SERVICES SOCIAL NETWORKS FOR THE DISCOVERY PROCESS

3.1 Introduction

As discussed in the previous chapter, discovering suitable services that meet the requirements of users is still a challenging problem because of the huge number of these services available on the web. In this chapter, we present our contribution that aims to help alleviate this problem using the platform of social networks. The key idea is to actively involve both users and web services in the discovery process, instead of focussing only on the web services as they appear on some registries, or even on the composition links connecting them. We argue that by merging users and web services in the same social network, the discovery process becomes easier and more efficient. Moreover, we analyze the influence that users social networks and web services social networks have on each other.

3.1.1 Web Service / User Interaction Scenario

To motivate the importance of relationships among the different nodes in a social network, we will use an example that reveals the roles a node can act upon in such a network.

Let us consider the scenario depicted in Figure 3.1. Assume a user $user_i$ who wants to travel and needs a web service to book a ticket. One option $user_i$ can take is to start exploring her social network to see if some of her friends had satisfactorily used a similar

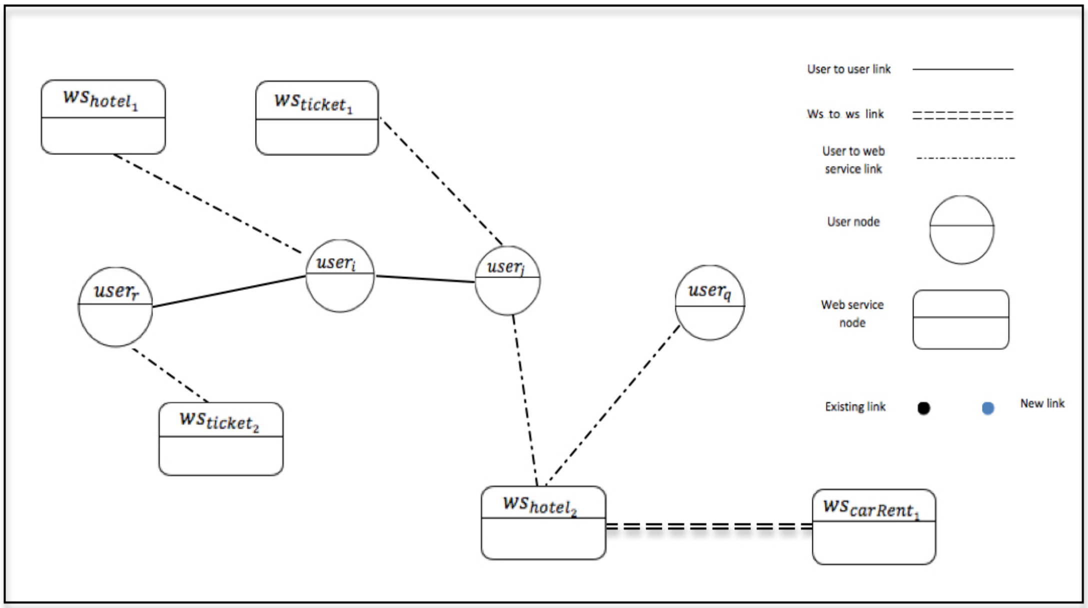

Figure 3.1 Initial state of the social network

service. Assume she successfully finds that two of her friends $user_j$ and $user_r$ have had used a similar service in the past $WS_{ticket1}$ and $WS_{ticket2}$ respectively. Then, $user_i$ will consider these web services, and after evaluating each one of them she will choose the suitable one for her needs. For instance, let $WS_{ticket1}$ be the chosen one. Moreover, let us assume that $user_i$ has a connection with another web service WS_{hotel1} because she had used this service to her satisfaction in the past. Thus, the two web services $WS_{ticket1}$ and WS_{hotel1} can be connected as well for a possible partnership in terms of composition in the future.

On the other hand, assume another user $user_q$ needs to rent a car, but she does not have any friend in her social network who have used this service. $user_q$ can explore her connections with web services to find that WS_{hotel2} has $WS_{CarRent}$ in its collaboration (or partnership) social network. Thus, $user_q$ can potentially use the service after evaluating it. In the same example, $user_q$ can also expand her social network through the web services that she had transactions with. If WS_{hotel2} has a connection with $user_j$ in addition to $user_q$, then these two users can add one another to their social networks.

Figure 3.2 shows the new topology of the considered social network after adding the connections discussed above. The idea we aim to motivate through this example is that users can discover new web services using their connections to other users or even web services, and web services can discover each other through shared users for potential business partnership. These two discovery aspects have not been investigated in the literature yet.

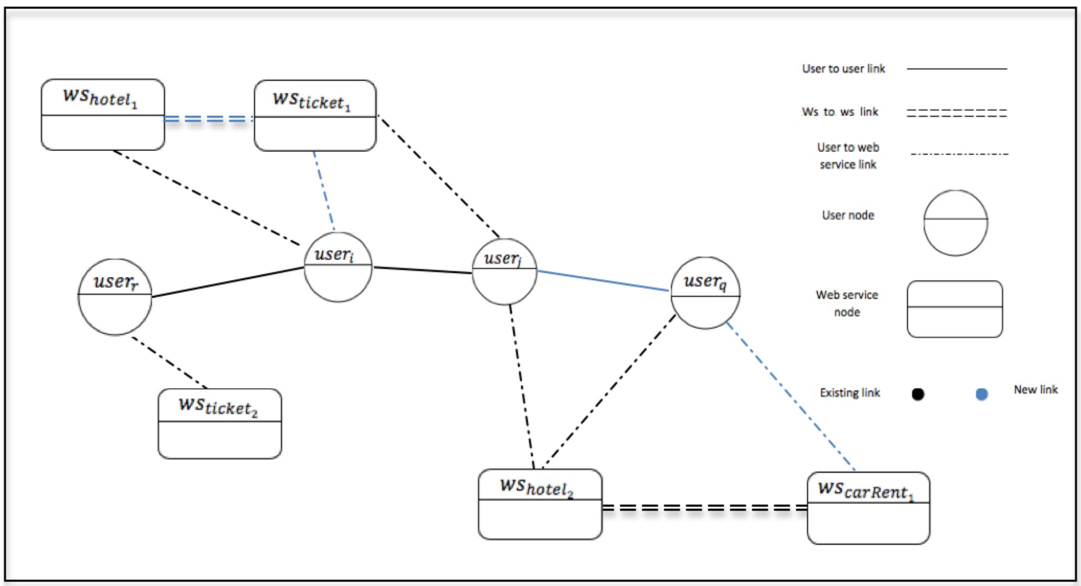

Figure 3.2 Status of the social network after adding connections

Thus, in the process of looking for a recommendation for a web service, various relationships and interactions could be captured. First, web services could expand their social networks by using the user as a medium. This is the case of $WS_{ticket1}$ and WS_{hotel1} with $user_i$ as a common user. Also, users can discover and add new peers to their social networks as well. For example, WS_{hotel2} has introduced $user_j$ and $user_q$ to each other. In addition, a user can discover a web service through her friends, like when $user_i$ found $WS_{ticket1}$ through $user_j$. The discovery can also be implemented through web services when a user can find a web service with the help of another web service. This is shown in the example when $user_q$ discovered $WS_{CarRent}$ via WS_{hotel2}.

3.2 Social Users and Web Services

3.2.1 Engineering Social Users and Web Services

Social networks have been used in different fields to help improve the human life. They are used as a good interactive learning tool [16, 23, 31, 42, 46], as a mean to exchange knowledge in the scientific communities [22, 24], and even in the enterprize and business context [34]. In fact, many scenarios have discussed users' social networks from different perspectives. In [15], the authors propose a generative model of a social network consisting of users. They take into their account two characteristics: the background, which is the group in the real word that the user belongs to, and the behavior, which illustrates how the user serves and interacts with other users. In another perspective, Cai and his colleagues have defined the social network nodes as active users and passive items [9]. They have also developed an algorithm called *SocialCollab* to predict which users are most likely to contact other users according to the similarity of both interests and taste.

In our work, we focus on building the users and web services social networks from an engineering perspective. Inspired by the steps used in [25], we use a three-step engineering method to help us construct our social network. However, unlike [25] that only focusses on the network of web services, in this thesis, we consider both, networks of users and those of web services in the same global network. With regard to the users social network, our approach consists of giving the users one role to play, which is to be a *Friend* with other users. For the web services social network, partnership in terms of composition and substitution is the key factor behind connecting different web services. Constructing these relationships will help us better understand users' interactions with each other and with other web services as well.

- **Step 1:** Establishing a relationship between users and web services.

  Generally, with respect to users, there are two ways to build a social relation [30]:

  - Explicit: The user offers the chance of forming a relationship with another specific user, i.e., two users become friends if one of them invites the other who accepts the invitation.
  - Implicit: Social relations can be deduced from the activity of the different users, i.e., a user uses many services of other users.

  In our users' social network, two users are friends if they had past interactions or they have some similar interests in terms of the services they usually invoke. On the other hand, two web services are linked if they had previous business transactions. Two transactions are considered: composition and substitution. Composition means that one of the web services completes the other to satisfy a complex user request. Composition between flight booking, hotel reservation, and car rental to satisfy a user request is a typical example. Substitution is a cooperation relationship [18], where a web service agrees to substitute another one, for instance if the first one is not available. Substitution can also be implemented using resource sharing among web services within the same community [19, 28].

- **Step 2** Identifying social network components.

  In the previous chapter, we discussed the two components of the web services social networks. Similarly, users social networks have two components: nodes which represent users, and edges which represent the relationship between those users. Accordingly, two types of information could be extracted from this network [30]: (1) user profile, which contains information about the user's particular interests and preferences, and the history of her interactions; and (2) links descriptions that define the social network. These links are used to calculate the social proximity between two users. From the social network of web services, functionalities and QoS provided are characterizing web service nodes.

To evaluate the weight of an edge between two users, we use the similarity level between these users SL_{u_i,u_j} in terms of invoked web services as follows:

$$SL_{u_i,u_j} = \begin{cases} 0 & \text{if } \; u_i \rightarrow ws \; \cup \; u_j \rightarrow ws = \emptyset \\ \frac{|u_i \rightarrow ws \; \cap \; u_j \rightarrow ws|}{|u_i \rightarrow ws \; \cup \; u_j \rightarrow ws|} & \text{otherwise} \end{cases} \tag{3.1}$$

where $u_x \rightarrow ws$ is the set of web services connected to the user u_x, which means the set of web services that have been invoked by that user in the past. The maximum value that this metric can take is 1, which means the two users are connected to exactly the same web services. This value measures how much a pair of users are similar by computing the number of web services that they have in common.

In the same way, we define the level of (market) similarity between two web services based on the number of common users:

$$SL_{ws_i,ws_j} = \begin{cases} 0 & \text{if } \; ws_i \rightarrow u \; \cup \; ws_j \rightarrow u = \emptyset \\ \frac{|ws_i \rightarrow u \; \cap \; ws_j \rightarrow u|}{|ws_i \rightarrow u \; \cup \; ws_j \rightarrow u|} & \text{otherwise} \end{cases} \tag{3.2}$$

where $ws_x \rightarrow u$ is the set of users connected to the web service ws_x.

- **Step 3** Identifying user and web service characteristics.

Different social characteristics of *user* and *web service* nodes can be considered. In this thesis, we focus on the following metrics.

Popularity. In a pure friends' social network, the popularity among friends of a user node is computed based on the number of users connected to that user as follows:

$$P_{u_i} = \frac{|u_i \rightarrow u|}{|U|} \tag{3.3}$$

where $u_i \rightarrow u$ is the set of user nodes connected to the user u_i and U is the set of user nodes in the considered social network.

Similarly, the popularity of a web service among its peers is computed as follows:

$$P_{ws_i} = \frac{|ws_i \rightarrow ws|}{|W|} \tag{3.4}$$

where $ws_i \rightarrow ws$ is the set of web service nodes connected to the service ws_i and W is the set of web service nodes in the considered social network.

Activity. In our social network that merges users and web services, the activity degree of a user is measured based on the number of web services that user had previously invoked:

$$A_{u_i} = \frac{|u_i \rightarrow ws|}{|W|} \tag{3.5}$$

Market Share. The market share of a particular web service ws_i is computed based on the number of users who invoked that service:

$$M_{ws_i} = \frac{|ws_i \rightarrow u|}{|U|} \tag{3.6}$$

Satisfaction. The degree of satisfaction $S_{u_j}^{u_i}$ a user u_j has about a user u_i is measured in terms of the distance between the recommendation $r_{u_i \rightarrow u_j}^k$ given by u_i and the actual evaluation $v_{u_j}^k$ done by u_j for each recommendation k. By recommendation, we mean the value a user assigns to a particular web service based on the QoS received:

$$S_{u_j}^{u_i} = e^{-\sum_{k=1}^{n} |r_{u_i \rightarrow u_j}^k - v_{u_j}^k|} \tag{3.7}$$

3.3 Web Service and User Discovery

After analyzing the construction of users and web services social network, we need to understand the type of interactions that could occur between these two components while they are working together. The different interactions that can be described between those two types of nodes using the social network paradigm fall in one of two categories: web services discovery, and users discovery.

3.3.1 Web Services Discovery Through Users' Social Network

The main role of users in our extended social network merging services and users is to give evaluation of web services to other peers in order to help them receive a good QoS and give incentive to web services to perform better since those evaluations are made public. Furthermore, users can also assist in the discovery process of web services by helping other users find web services, and helping web services themselves extend their social connections to other users and web services.

A. First Case

When a user u_j starts searching for a certain service, first she will look for user friends who have used a similar service. In the ideal situation, u_j will find some users who made their evaluations public of one or many services similar (in terms of functionality) to the one she is looking for. Figure 3.3 shows the simple example of u_j that discovers ws_q through u_i. The user u_j should evaluate ws_q based on her own evaluations of her friends and how those friends evaluate ws_q. This evaluation is given in Equation 3.8 where:

- $f_{u_j}^{ws_q}$ is the set of u_j friends that have made their evaluations of ws_q public.
- $DW_{u_i}^{ws_q}$ is the direct evaluation u_i has about the service ws_q.

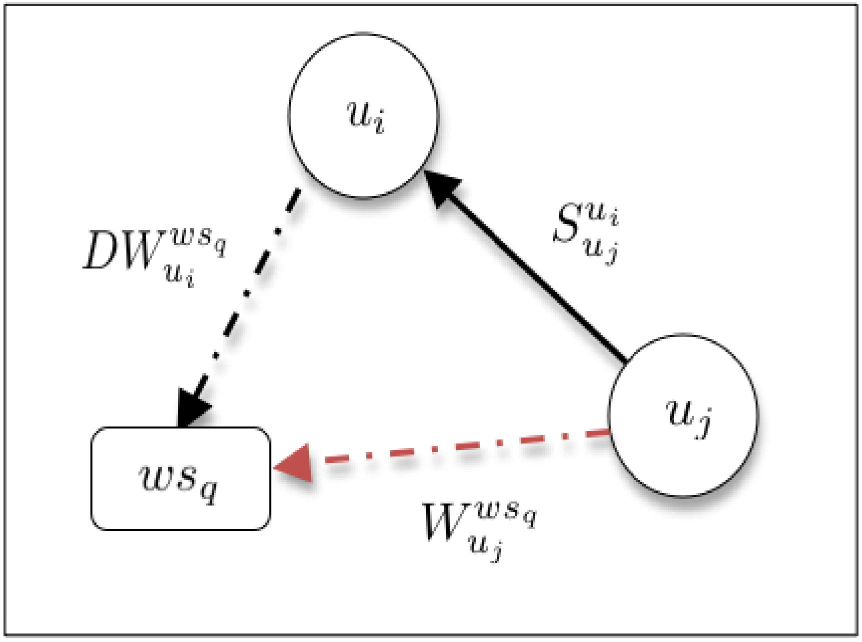

Figure 3.3 Discovery of a web service by a user through another user

$$W_{u_j}^{ws_q} = \frac{\sum_{u_i \in f_{u_j}^{ws_q}} (S_{u_j}^{u_i} \times DW_{u_i}^{ws_q})}{\sum_{u_i \in f_{u_j}^{ws_q}} S_{u_j}^{u_i}} \tag{3.8}$$

The direct evaluation that a user u_i has about a web service ws_q is computed after a number n of requests u_i sends to ws_q using the requested QoS for the interaction k ($RQ_{u_i}^{ws_q}(k)$) and the obtained one ($QoS_{u_i}^{ws_q}(k)$):

$$DW_{u_i}^{ws_q} = \sum_{k=1}^{n} \frac{RQ_{u_i}^{ws_q}(k)}{QoS_{u_i}^{ws_q}(k)} \tag{3.9}$$

B. Second Case

From another point of view, one user can help two web services discover each other if that user has used both of them and those web services never had a connection before (Figure 3.4). This discovery can be used to build a new partnership relation. Such a

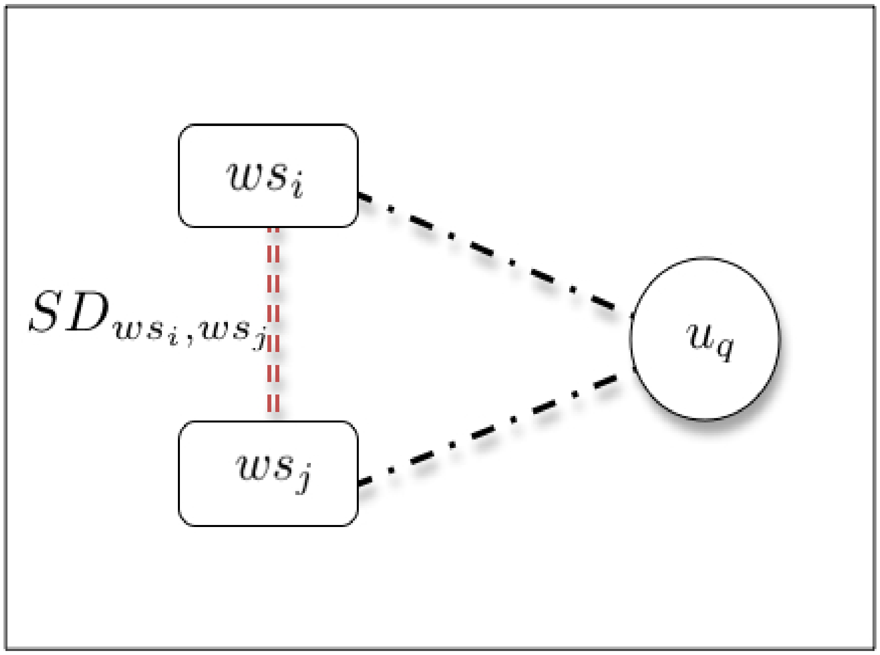

Figure 3.4 Discovery of a web service by another web service through a common user

relation can be either substitution if the two web services are functionally similar, or composition, if they are complementary. Hereafter, we will discuss these two relations.

- If the two web services have similar functionalities, substitution can take place if the degree of similarity is high enough. Such a degree should take into account the profile of the two web services as argued in [27]. The profile of a web service has five categories: preconditions, inputs, outputs, effects and QoS. Preconditions are elements needed by the service prior to its execution, and effects are the expected outcomes that result from the execution of the service. For example, a service that sells items on the web may require as a precondition a valid credit card or a paypal account. The credit card number and expiration date or the paypal number and balance are examples of the required inputs. The generated receipt is the output. As effect, the card is charged or the paypal account is updated. The QoS can be expressed in terms of different metrics such as response time, throughput, availability, reliability, etc. To compute the similarity degree SD_{ws_i,ws_j} between the two web services ws_i and ws_j, we use the *ontological matching score* between concepts describing the web services profiles as proposed in [27]. However, unlike [27] that uses this score to compare all the concepts, we compare the concepts belonging to the same category. By doing so, we obtain a more accurate similarity degree as only related concepts are used. Equation 3.10 computes this metric:

$$SD_{ws_i,ws_j} = \frac{\sum_e w_e \frac{\sum_k w_k^e \times MS(c_{i,k}^e, c_{j,k}^e)}{\sum_k w_k^e}}{\sum_e w_e} \tag{3.10}$$

where:

– The index e is used for the category and the index k for the similar concepts within the same category.

– $MS(c_{i,k}^e, c_{j,k}^e)$ is the matching score of a pair of concepts c in two web services ws_i and ws_j [27].

– w_e is the weight of the category and w_k^e is the weight of the matching score within the same category.

The main idea is that a web service ws_i will add ws_j to its substitution social network if the degree of similarity SD_{ws_i,ws_j} is greater than a given threshold.

- If the two web services are complementary, then they can be connected for a potential composition. Two web services ws_i and ws_j are complementary if the inputs of one match the outputs of the other. The same equation used to compute the similarity degree (Equation 3.10) can be used to compute the the complementarity degree CD_{ws_i,ws_j} by restricting the concepts to the input and output categories, where the inputs concepts are compared to the output ones.

3.3.2 User Discovery Through Web Services Social Network

In the literature, the discovery process is always initiated by the user who is supposed to look for services. However, in the modern approaches of economics, services are also active entities and can introduce themselves to users. In our framework, we support this introduction using the concept of users and web services social network. For instance, in the first case of the previous section (see Figure 3.3), not only a user can discover a web service through another user, but also a web service can discover a user through another user.

When peers in a web service social network interact with each other, usually they collaborate, compete or substitute each other. However, they can have different types of communication when they exchange information about users' trust [17]. We will discuss two cases that will help us understand the process of discovering and sharing user information.

A. First Case

Consider a user u_q who has a complex request that needs more than one web service to be satisfied. That user can utilize its connection to an initial web service ws_i that could recommend another web service ws_j. This situation is very similar to the first case in the previous section. However, the same situation can be seen from another perspective, where the web service ws_i that is connected to the user u_q recommends that user to the web service ws_j (see Figure 3.5). This will happen if ws_i is collaborative and a business partnership is already established between ws_i and ws_j. The establishment of the connection between the web service ws_j and the user u_q will depend on how much ws_j trusts ws_i ($DT^{ws_i}_{ws_j}$), and how much ws_i trusts u_q ($DT^{u_q}_{ws_i}$). The case can be generalized when u_q is recommended by many other web services. Equation 3.11 shows how ws_j evaluates the trust of u_q using information conveyed from recommenders of u_q that are business partners of ws_j ($b^{u_q}_{ws_j}$). This trust is the initial weight at time t of the transition

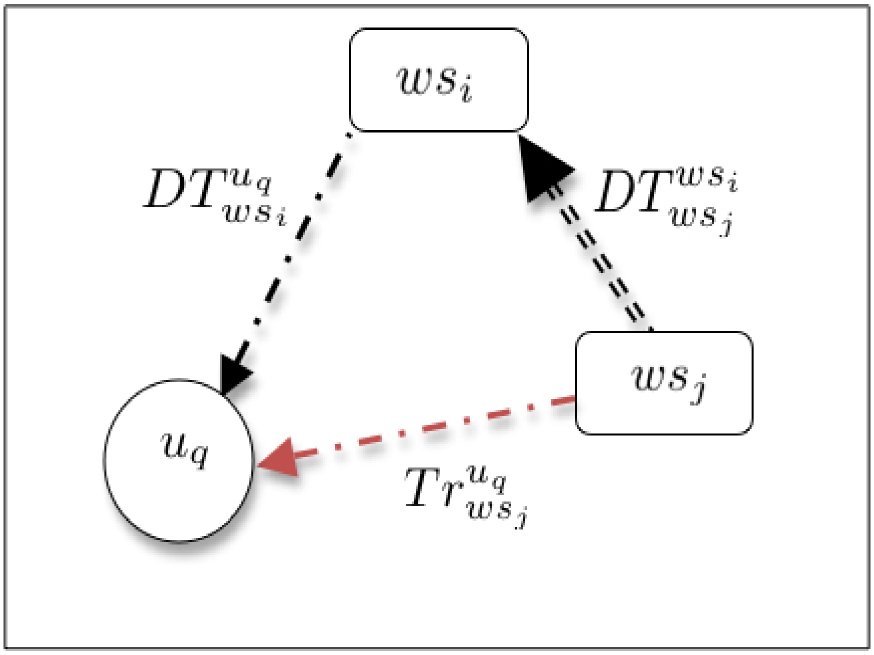

Figure 3.5 Discovery of a user by a web service through web service

linking the web service to the user ($weight_t(ws_j, u_q)$).

$$Tr^{u_q}_{ws_j} = \frac{\sum_{ws_i \in b^{u_q}_{ws_j}} (DT^{ws_i}_{ws_j} \times DT^{u_q}_{ws_i})}{\sum_{ws_i \in b^{u_q}_{ws_j}} DT^{ws_i}_{ws_j}} \tag{3.11}$$

The direct trust $DT^{ws_i}_{ws_j}$ is computed as follows:

$$DTr^{ws_i}_{ws_j} = \frac{\sum_{l=1}^{n} (\lambda_l . TiR^{ws_i}_{ws_j} . r_l)}{\sum_{l=1}^{n} (\lambda_l . TiR^{ws_i}_{ws_j})} \tag{3.12}$$

where r_l is the rating that ws_j gave ws_i for the past transaction l, λ_l is the importance of that interaction, $TiR^{ws_i}_{ws_j}$ its time recency, and n is the number of the total transactions between ws_j and ws_i. The direct trust $DT^{u_q}_{ws_j}$ is computed in the same way.

The web service ws_j will update its weight with the user u_q as follows:

$$weight_{t+\delta t}(ws_j, u_q) = weight_t(ws_j, u_q) + \alpha \times [DTr^{u_q}_{ws_j} - Weight_t(ws_j, u_q)] \tag{3.13}$$

where α is a constant between 0 and 1, and δt is the update period.

Also as stated in [14], the weight between ws_i and ws_j will be updated as follows:

$$weight_{t+\delta t}(ws_i, ws_j) = weight_t(ws_i, ws_j) + \alpha \times [\frac{|ws_jSelection|}{|ws_iCollaboration|} - weight_t(ws_i, ws_j)] \quad (3.14)$$

where $|ws_iCollaboration|$ is the number that ws_i has participated in a composition, $|ws_jSelection|$ is the number of times that ws_j has been selected by ws_i from other peers to participate in a composition.

B. Second Case

In some scenarios, one web service can serve two users from two different social networks. If the two users are not linked, they can discover each other through the common web service as shown in Figure 3.6. In fact, the web service that the two users are sharing can recommend them to build a connection. Motivated by the fact that there is a big chance that both users have similar needs so they may use similar services, we use the Similarity Level (Equation 3.1) as the weight. To update the two edges connecting the web service with the two users, we use Equation 3.13.

3.4 Conclusion

In this chapter, we proposed a new framework for the discovery process based on merging users and web services social networks in the same global network. We showed the benefits of merging these two social networks to assist users and web services discover other peers. We have presented an interaction scenario that sums up how web services and users can expand their networks. Then, we focused on building a social network for users and web services and identified the potential behaviors that users and web services can accept inside their networks. We discussed four different scenarios that

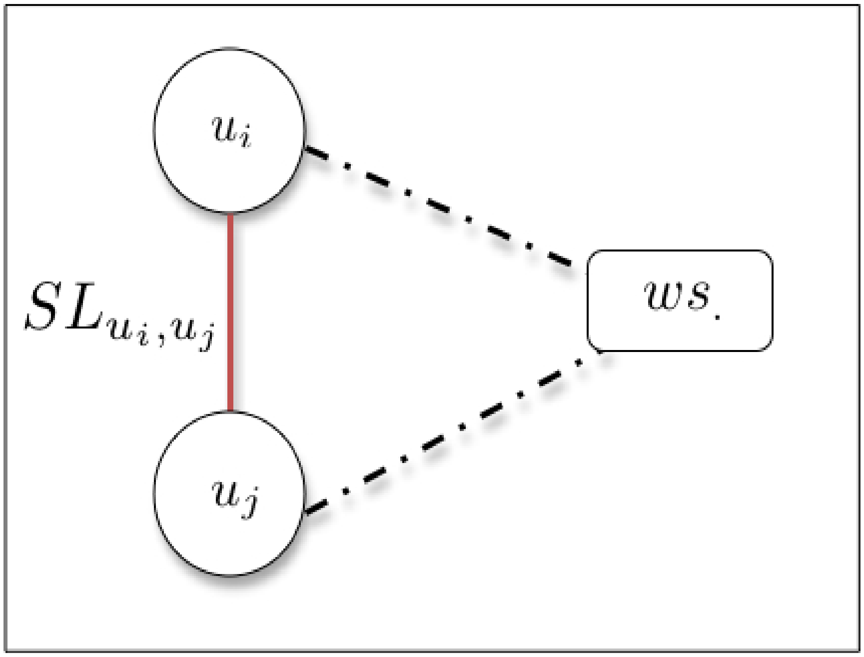

Figure 3.6 Discovery of users that share a web service

could occur during the discovery process. An important aspect of our framework, is that unlike existing solutions, we not only help users discover web services, but also allow web services introduce themselves to potential users. In the next chapter, we will present our implementation and simulation process with the result analysis.

CHAPTER 4. SIMULATION AND RESULTS ANALYSIS

In this chapter, we will examine our simulation process that will illustrate the value of the proposed framework. Thereafter, we will analyze the results that we obtained.

4.1 Simulation

In the simulation process, we built two social networks, one for web services and one for users. These social networks started with few connections among one another. What we will be able to see over time is how these networks are able to evolve and grow using only the initial connections that they started with. By achieving this, web services and users will be able to find new peers that meet their requirements, without having to check the UDDI registry each time they want to find a new service.

The simulation is written in Java in the Eclipse environment. We simulated two social networks with 100 nodes each, which represent web services and users. These nodes are connected through three types of links according to the node type. We list the variable that we used in Table 4.1. The "Value" column illustrates the values that we used in our experiment, and all these values can be changed by the evaluator.

We compared our discovery approach to a recent approach that uses only web services' social networks for discovery [25, 27, 29]. This approach does not consider users as a factor that can help web services expand their social connections with other peers, or as a way of advertising among new users. In the simulation process, we built two social networks, one for web services and the other for users. We set the number of nodes to 100

Parmeters	Value
Number of web services	100
Number of web service types	2
Number of users	100
Number of iterations	1 to 100
Two web services chance to initially get connected	30%
Web service and user chance to initially get connected	30%
Two users chance to initially get connected	30%

Table 4.1 Social Network Parameters

in each network so that we can observe the changes easily. Each node can be connected through three types of links: User - User link, Web Service - Web Service link, and Web Service - User link. Each one of these connections has a certain weight to be calculated as mentioned previously in Chapter 3. The social network will have few connections between the nodes at the initial state, and as the users nodes start requesting services the connections of the graph will progress fast, not only between users, but among web services as well. According to our proposed framework, the discovery process will be governed by one of four scenarios. Either a user will help another peer to find the service that she needs, or a user will assist a web service to expand its social network by adding new web service after a certain number of interactions. Other scenarios show how web services can introduce their peers to new users, or work as a web service discovery tool for users.

4.2 Results and Analysis

In this section, we aim to prove that by integrating users' social networks into the web services discovery, we are able to reach more users for advertising, and improve the possibility of expanding the web services social network.

4.2.1 Effect of Users' Social Network on Edge Formation

According to our scenario, three types of edges will be formed between nodes:

A- Edge between two Web Services

This type of edge is used to expand web services' social network. In fact, the growth of the number of links between web services gives us a good indicator of the graph expansion, which makes the discovery easier. According to our approach, this type of edge is formed when one user uses services from two different web services more than twice, so this user will introduce these web services to each other. Figure 4.1 shows that the number of connections using social users will make the graph grow faster compared to the approach where only web services are socially connected. Consequently, in our approach, more peers are getting discovered.

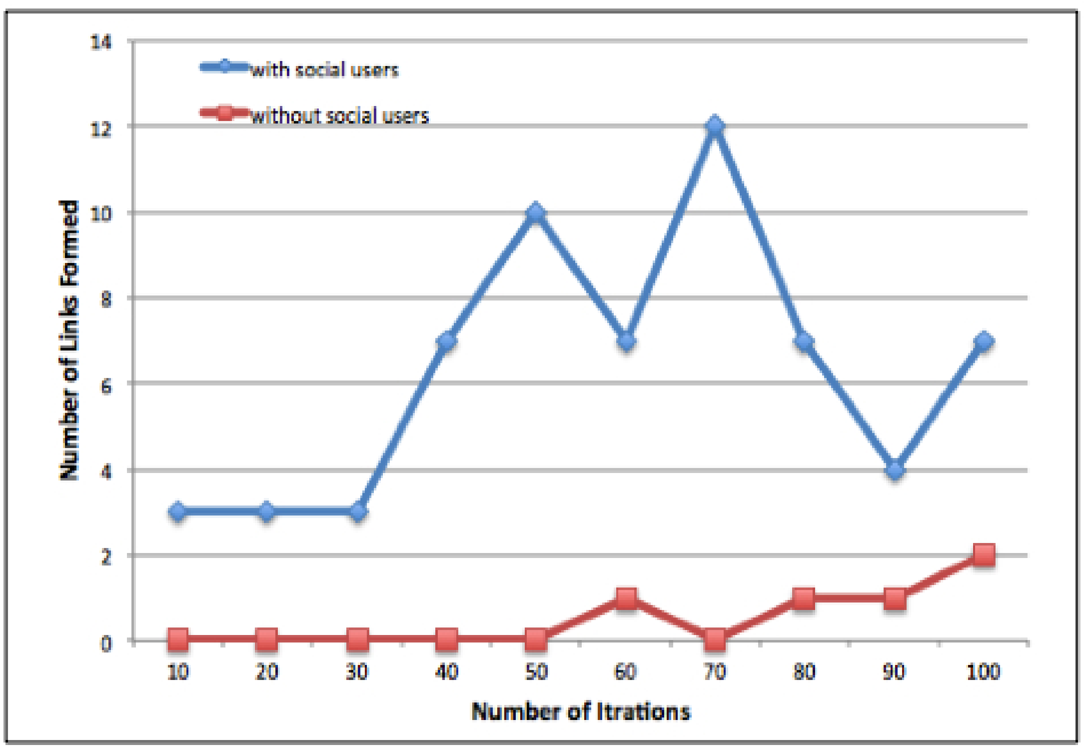

Figure 4.1 Web Service - Web Service edge formation

B- Edge between Web Service and User

This type of link will help web services reach a wider group of users to advertise their services by making use of the information captured using the global social network. Without using users' social network, web services tend to wait for the costumer request from the provider after searching the UDDI for that service. However, when our method is deployed, users can be involved in the process of finding the appropriate service using their connections to the other users in their social network. On the other hand, web services also can perform the same task as users, which means they can find a web service that can provide the needed task when requested. Figure 4.2 illustrates the number of connections that are formed when our approach is used and compares it to the number of links formation if only the social network of web services is used, but no social user is involved.

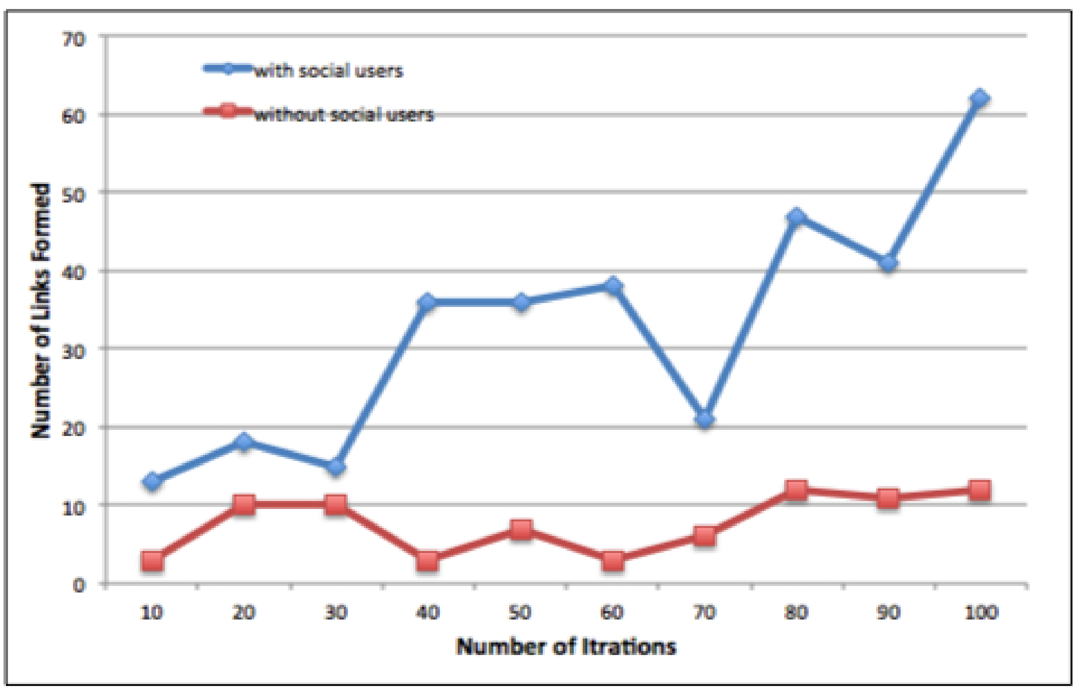

Figure 4.2 Web Services - User edge formation

C- Edge between two users

The importance of this type of edge is to form and expand users' social networks. As shown in the previous chapter, this social network can effectively help web services find both other web services and users. To form this link, a web service will recommend two users to get connected, if that web service already has connections with those users and they both have been using his service more than twice. Figure 4.3 shows the relation between users' numbers in their social network and the number of edges formed among web services and among users and web services.

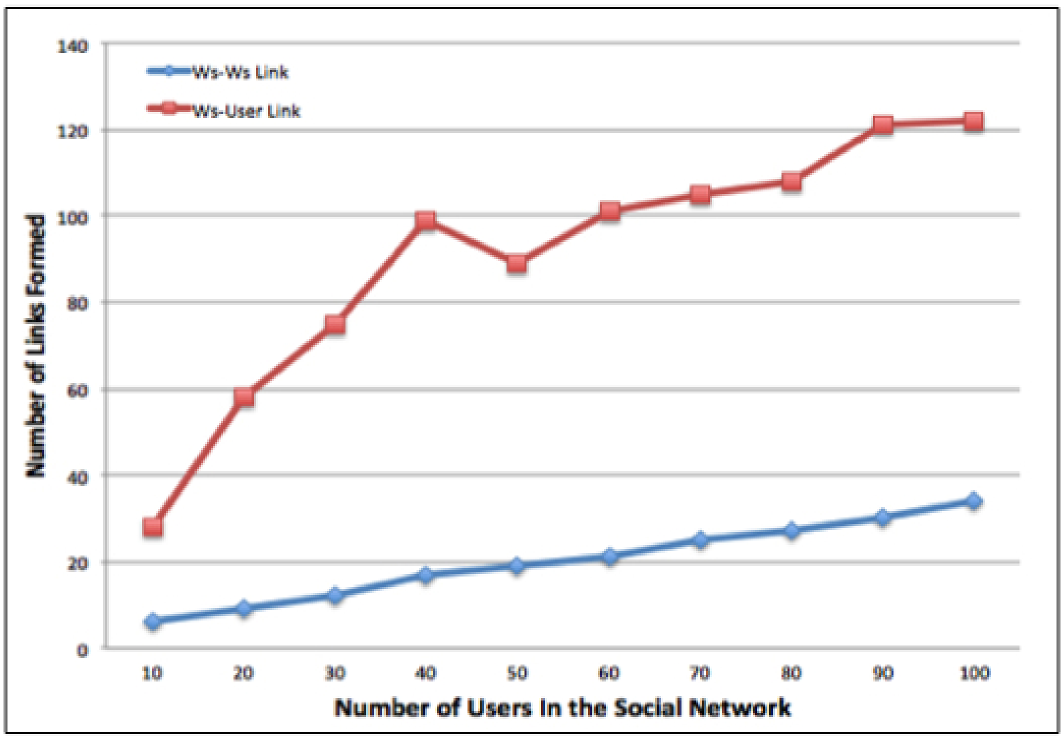

Figure 4.3 Effect of the number of users on the number of edges formed

4.2.2 Effect of User behavior on Social Networks

In this section, we will analyze users' popularity and selfish behavior and their impact on users social network and web services advertising.

A- Popular User

As we mentioned in Chapter 3, a user can be popular in his social network if he has the largest number of links to users in his social network. As shown in Figure 4.4, we found that popular users tend to connect with fewer web services compared to other users. The reason is that they have a sizeable number of connections (or friends) which makes it easer to find a quality web service than for those who have a fewer friends. In other words, the probability of finding good web services increases if the user has a wider access to other users in the network.

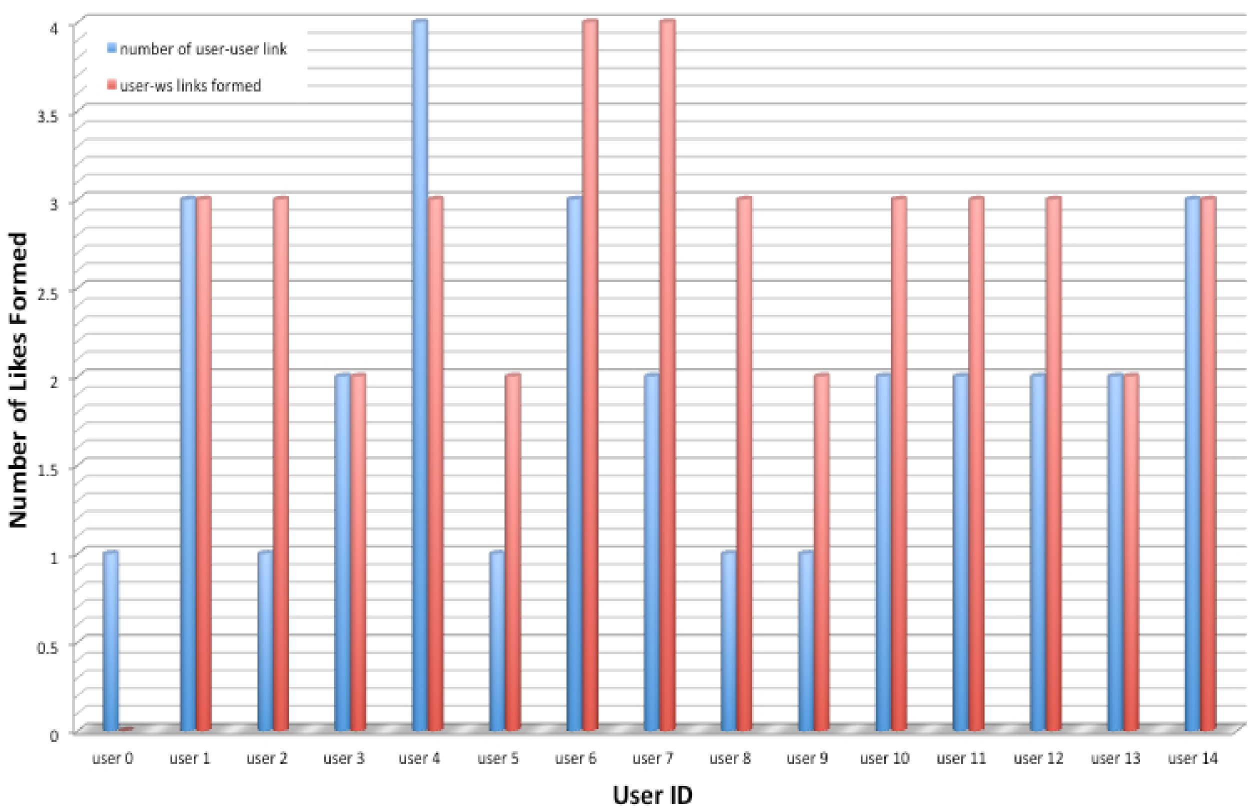

Figure 4.4 Relation between popular users and User-Web Service edge formation

4.2.3 Selfish User

A user behaves in a selfish manner if he refuses to disclose his links or evaluations to other peers. In Figure 4.6, we illustrate how selfish behavior can impact negatively the ability of users to connect with new peers in the social network. This could be related to their location on the graph, in which they do not have a lot of users around them, or because they have already connected to high quality web services so they don't have to consult with other users.

However, there is no significant association between forming new links with new web services and having a small number of interactions with other users as shown in Figure 4.5. Yet, this behavior can affect the advertisement of web services, in which fewer connections for a selfish user means less exposure for the web services that the user connected to.

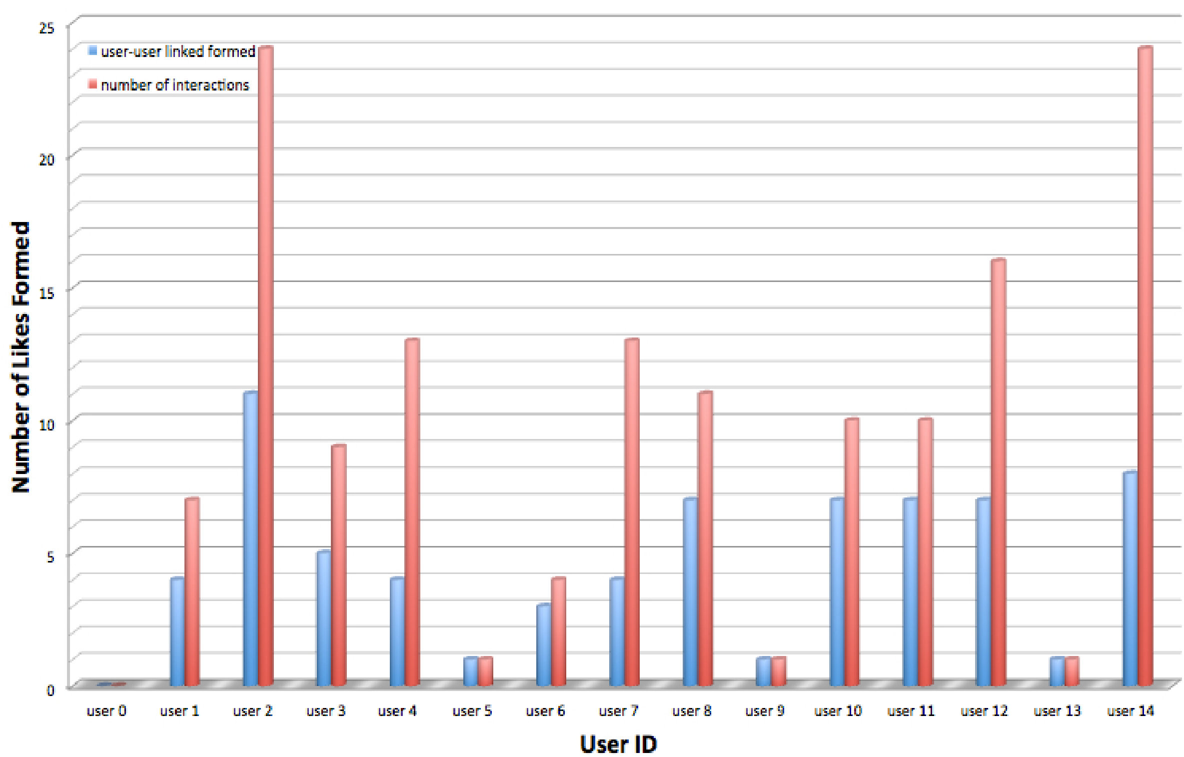

Figure 4.5 Relationship between selfish users and User - User edge formation

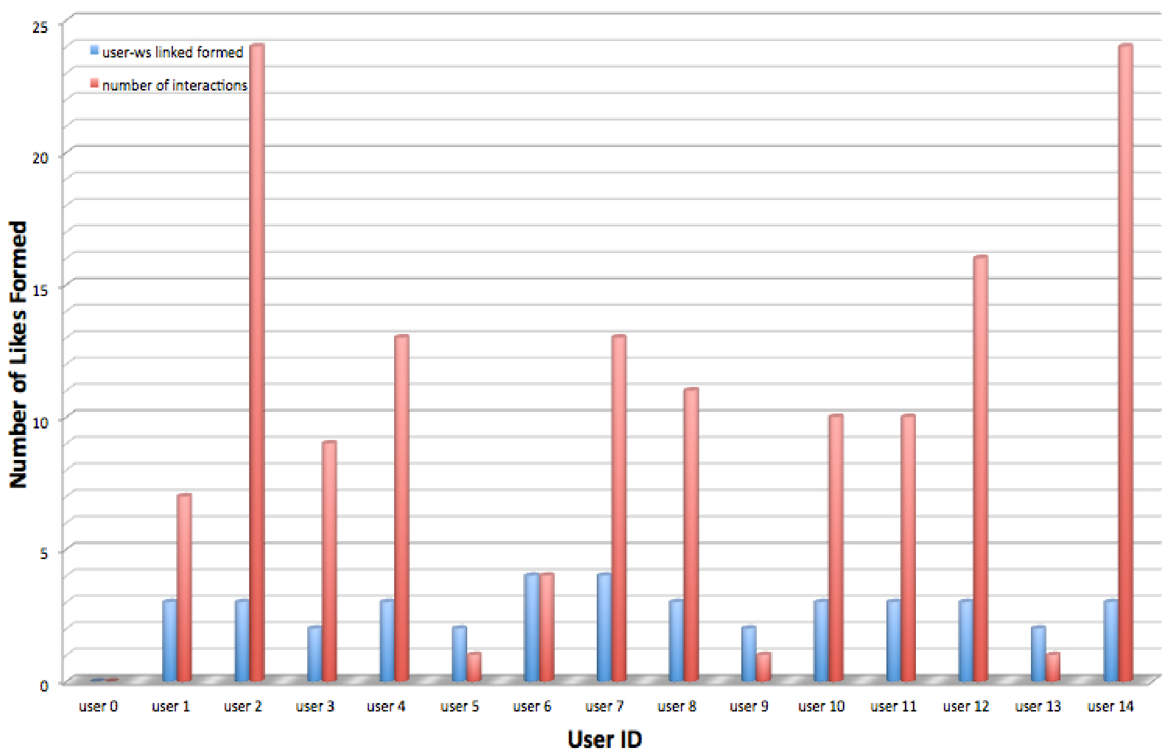

Figure 4.6 Relation between selfish users and User - Web Service edge formation

4.3 Conclusion

In this chapter, we presented our simulation results and showed the effectiveness of our approach. The use of users' social networks has demonstrated significant improvement in discovering new web services and advertising for more users. As the number of links between web services represent the discovery process, our method did very well against the outer method that uses only the web services social network. Moreover, when we compared the number of Web Service - User edges formed, we found that our process did much better than the other method. Moreover, we found that by increasing the number of users in a social network, we also increased the chances of discovering more web services and advertising to more users. on the other hand, when we studied the relationship between user behaviors and social networks, we found that popular users usually connected to high QoS web services as they could find them faster than other users due to their higher number of connections. Also, we discovered that users with a smaller number of interactions do not form a large number of links with other users. However, this does not affect the user's ability to find web services.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Summary of Contributions

In this thesis, we proposed a new framework of discovery for web services. Our method has introduced the idea of using social users to enhance advertisement and discovery of web services. The incorporation of users and web services social networks helps solve many problems that face web services, such as finding the best location for advertisements and replacing services in case of failure. Also, adding the social aspect makes the user more involved in discovering the appropriate web services he is looking for, and at the same time involves him in the process of expanding his web service social network.

To integrate users and web services in the same social network, we have used a three-step engineering process to achieve our goal: establishing a relationship between users and web services, incorporating this relationship in to the social network, and identifying user and web service characteristics. Then, we presented four different scenarios that can occur while building our approach. These different scenarios resulted in forming three types of links: user to user edges, web services to user edges, and web services to web services edge. Each one has it's own weight formulas.

We have compered our method against another method, which does not use social users, but focuses only on the link between services to prove our method's effectiveness. The results showed a significant improvement in the advertisement and discovery of web services. Furthermore, we have observed the effects of users' behaviors in the ability to expand their connections inside their social network and in web services social network.

5.2 Future Work

Many research opportunities have been acknowledged to promote the work that has been done. Some of these suggestions are listed below:

- Study the connection between the social network shape and the discovery process. The architecture of the social network and the position of the nodes inside the network can play a major role in the navigation mechanisms.
- Apply social network analysis (e.g., clustering coefficient, centrality, distribution, segmentation, etc.) on our global web services and users social networks, and than analyze their connections to web services and users discovery and web services advertisement.
- Include a trust assessment approach that can be applied to social web services as well as users.

BIBLIOGRAPHY

[1] Sudhir Agarwal and Rudi Studer. Automatic matchmaking of web services. In *Web Services, 2006. ICWS'06. International Conference on*, pages 45–54. IEEE, 2006.

[2] RM Araujo, Erick A Rezende, Tiago S Andrade, Victor Manaia Chaves, Mauro G Lopes, and Bruna Diirr. People in network, collaboration for action: New supporting requirements. In *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, pages 939–944. IEEE, 2007.

[3] Albert-László Barabási and Jennifer Frangos. *Linked: The New Science Of Networks Science Of Networks*. Basic Books, 2002.

[4] M Bichier and K-J Lin. Service-oriented computing. *Computer*, 39(3):99–101, 2006.

[5] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. *Web Services Architecture*.

[6] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. *Simple Object Access Protocol (SOAP) 1.1*.

[7] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*.

[8] John Breslin and Stefan Decker. The future of social networks on the internet: the need for semantics. *IEEE Internet Computing*, 11(6):86–90, 2007.

[9] Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke, Yang Sok Kim, Paul Compton, and Ashesh Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *AI 2010: Advances in Artificial Intelligence*, pages 476–485. Springer, 2011.

[10] Francisco Christensen, Erik andCurbera, Greg Meredith, and Sanjiva Weerawarana. *Web Services Description Language (WSDL) 1.1.*

[11] Beniamino Di Martino. Semantic web services discovery based on structural ontology matching. *International Journal of Web and Grid Services*, 5(1):46–65, 2009.

[12] David Easley and Jon Kleinberg. *Networks, crowds, and markets*, volume 8. Cambridge Univ Press, 2010.

[13] May El Barachi. *Using Web services for application development in internet telephony: a case study on conferencing in SIP networks.* PhD thesis, Concordia University, 2004.

[14] Noura Faci, Zakaria Maamar, and Parisa Ghodous. Which social networks should web services sign-up in? In *AAAI Spring Symposium: Intelligent Web Services Meet Social Computing*, volume SS-12-04 of *AAAI Technical Report*. AAAI, 2012.

[15] Ilias Foudalis, Kamal Jain, Christos Papadimitriou, and Martha Sideri. Modeling social networks through user background and behavior. In *Algorithms and Models for the Web Graph*, pages 85–102. Springer, 2011.

[16] Li Jin and Zhigang Wen. An augmented social interactive learning approach through web2. 0. In *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International*, volume 1, pages 607–611. IEEE, 2009.

[17] Babak Khosravifar, Jamal Bentahar, and Maziar Gomrokchi. Declarative and numerical analysis of edge creation process in trust-based social networks. In *Declarative Agent Languages and Technologies VII*, pages 137–157. Springer, 2010.

[18] Babak Khosravifar, Jamal Bentahar, Rabeb Mizouni, Hadi Otrok, Mahsa Alishahi, and Philippe Thiran. Agent-based game-theoretic model for collaborative web services: Decision making analysis. *Expert Syst. Appl.*, 40(8):3207–3219, 2013.

[19] Ehsan Khosrowshahi-Asl, Jamal Bentahar, Hadi Otrok, and Rabeb Mizouni. Efficient coalition formation for web services. In *IEEE SCC*, pages 737–744, 2013.

[20] Dirk Krafzig, Karl Banke, and Dirk Slama. *Enterprise SOA: service-oriented architecture best practices.* Prentice Hall Professional, 2005.

[21] Domenico Laforenza, Franco Maria Nardini, and Fabrizio Silvestri. Collaborative ranking of grid-enabled workflow service providers. In *Proceedings of the 17th international symposium on High performance distributed computing*, pages 227–228. ACM, 2008.

[22] Julia Porter Liebeskind, Amalya Lumerman Oliver, Lynne Zucker, and Marilynn Brewer. Social networks, learning, and flexibility: Sourcing scientific knowledge in new biotechnology firms. *Organization science*, 7(4):428–443, 1996.

[23] Lori Lockyer and John Patterson. Integrating social networking technologies in education: a case study of a formal learning environment. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 529–533. IEEE, 2008.

[24] John S Luo. Social networking: now professionally ready. *Primary Psychiatry*, 14(2):21–24, 2007.

[25] Zakaria Maamar, Noura Faci, Leandro Krug Wives, Hamdi Yahyaoui, and Hakim Hacid. Towards a method for engineering social web services. In *Engineering Methods in the Service-Oriented Context*, pages 153–167. Springer, 2011.

[26] Zakaria Maamar, Hakim Hacid, and Michael N Huhns. Why web services need social networks. *IEEE Internet Computing*, 15(2):90–94, 2011.

[27] Zakaria Maamar, Pedro Santos, Leandro Wives, Youakim Badr, Nora Faci, and José Palazzo M de Oliveira. Using social networks for web services discovery. *IEEE Internet Computing*, 15(4):48–54, 2011.

[28] Zakaria Maamar, Philippe Thiran, and Jamal Bentahar. Web services communities: From intra-community coopetition to inter-community competition. *Web Services Communities. E-Business Applications for Product Development and Competitive Growth: Emerging Technologies. Ed. In Lee. Hershey: IGI Global*, pages 333–343, 2011.

[29] Zakaria Maamar, Leandro Krug Wives, Youakim Badr, Said Elnaffar, Khouloud Boukadi, and Noura Faci. Linkedws: A novel web services discovery model based on the metaphor of "social networks". *Simulation Modelling Practice and Theory*, 19(1):121–132, 2011.

[30] Abderrahmane Maaradji, Hakim Hacid, Johann Daigremont, and Noël Crespi. Towards a social network based approach for services composition. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[31] Victoria J Marsick and Karen E Watkins. Informal and incidental learning. *New directions for adult and continuing education*, 2001(89):25–34, 2001.

[32] David W McDonald. Recommending collaboration with social networks: a comparative evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–600. ACM, 2003.

[33] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[34] Jody Nimetz. Jody nimetz on emerging trends in b2b social networking. *Marketing Jive (November 18, 2007)*, 2007.

[35] Tim O'reilly. What is web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*, page 17, 2007.

[36] Mike P Papazoglou. Service-oriented computing: Concepts, characteristics and directions. In *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 3–12. IEEE, 2003.

[37] Manoj Parameswaran and Andrew B Whinston. Social computing: An overview1. *Communications of the Association for Information Systems (Volume 19, 2007)*, 762(780):762, 2007.

[38] Josep M Pujol, Ramon Sangüesa, and Jordi Delgado. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 467–474. ACM, 2002.

[39] R Rajmohan, N Padmapriya, and SKV Jayakumar. A survey on problems in distributed UDDI. *International Journal of Computer Applications*, 36(3), 2011.

[40] Atul Sajjanhar, Jingyu Hou, and Yanchun Zhang. Algorithm for web services matching. In *Advanced Web Technologies and Applications*, pages 665–670. Springer, 2004.

[41] Munindar P Singh and Michael N Huhns. *Service-oriented computing: semantics, processes, agents*. Wiley. com, 2006.

[42] Peter A Smith. *Social networks and social networking*. Emerald Group Pub., 2009.

[43] Sebastian Stein, Katja Barchewitz, and Marwane El Kharbili. Enabling business experts to discover web services for business process automation. In *Emerging Web Services Technology, Volume II*, pages 23–39. Springer, 2008.

[44] uudi. *Online community for the Universal Description, Discovery, and Integration OASIS Standard*.

[45] Fei-Yue Wang, Kathleen M Carley, Daniel Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83, 2007.

[46] Bram De Wever, Peter Mechant, Pieter Veevaete, and Laurence Hauttekeete. E-learning 2.0: social software for educational use. In *Multimedia Workshops, 2007. ISMW'07. Ninth IEEE International Symposium on*, pages 511–516. IEEE, 2007.

[47] wikipedia. *Social Network*.

[48] Hamdi Yahyaoui, Zakaria Maamar, Erbin Lim, and Philippe Thiran. Towards a community-based, social network-driven framework for web services management. *Future Generation Comp. Syst.*, 29(6):1363–1377, 2013.

[49] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[50] Jun Zhang and Mark S Ackerman. Searching for expertise in social networks: a simulation of potential strategies. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 71–80. ACM, 2005.