Towards Agent-based and Trust-oriented Service Discovery Approach in Social Networks

Amine Louati Joyce El Haddad Suzanne Pinson

PSL, Université Paris-Dauphine, CNRS LAMSADE UMR 7243, France {amine.louati,elhaddad,pinson}@lamsade.dauphine.fr

Abstract

Service discovery has been adopted in distributed environments and recently in *social networks*. Existing approaches for service discovery are often based on matching techniques, which only capture common quality of service (QoS) criteria while ignoring the contribution of the social aspect. The discovery problem increases when the number of available services is important and no means to distinguish between two or many providers offering the same service. To overcome this issue, we propose a *trust measure* defined as the combination of two dimensions namely trust in sociability and trust in expertise. The problem is growing when no central control can be fixed due to the distributed nature of social networks. To cope with centrality related issues, we propose an agent-based service discovery approach in which each user in the social network is assigned an autonomous and non malicious agent equipped with a bounded set of services with their advertised QoS values. Endowed with a limited view of the social network, the agent acts locally on behalf of its user to discover trustworthy providers with good services. The search propagation process within the social network is ensured by means of a distributed referral system where agents cooperate and evaluate referrals based on a distributed knowledge and a decentralized decision-making.

Keywords: Service Discovery, Trust, Multi-Agent Systems, Social Networks, Referral Systems

1 Introduction

Current research in service discovery (see [PKPS02, SPAS03, RKM04] for example) are based on semantic and/or structural matching techniques between service requester needs and web services description i.e., functional and non-functional properties and follow a centralized service registries used by Universal Description, Discovery, and Integration (UDDI). However, with the emergence and rapid proliferation of media and social applications especially social networks (SNs), agents show the willingness to use their SNs to find services as well as to share services with others. Some studies [GA05, $MSW^+ 11$] have shown that SNs acts as a rich source for gathering information and services. Then, if we wish to rely on SNs for service discovery, we need to figure out how to capitalize on retrieved information to determine trustworthy providers with good services. Due to its limited view, a single agent is not able to discover relevant providers locally, and it needs some assistance from its

Copyright © by the paper's authors. Copying permitted only for private and academic purposes.

In: R. Cohen, R. Falcone and T. J. Norman (eds.): Proceedings of the 17th International Workshop on Trust in Agent Societies, Paris, France, 05-MAY-2014, published at http://ceur-ws.org

acquaintances or acquaintances' acquaintances to locate them i.e., analyzing SN structure. Agents interact by exchanging messages while evaluating the nature of the relationship and follow the relevant path to find trustworthy providers. The theory of Six Degrees of Separation claims that it is possible to reach anyone through a chain of acquaintances that has no more than five intermediaries. Humans are particularly good for finding these short paths [SJ05]. However, it is not that obvious in practice and especially in distributed environments such as SNs. Previous works [YS03, MHDC10, BBB10] don't consider the semantic aspect of relationships between agents and deal with a single-relation social network. However, the present work considers different relationship types between agents such as friendship, family relationship or, business relationship. Thus, our service discovery approach is established in a Multi-Relation Social Network (MRSN). In this paper we propose an agent-based and trust-oriented service discovery approach using a distributed referral system in multi-relation social networks that addresses the following challenges.

First, one of our main concerns in service discovery is the trust aware. A requester agent prefers commonly providers that not only propose needed services but also that can be trusted before using their services. If the requester knows that discovered services are offered by trustworthy providers, he will be more confident. For instance, when searching for babysitting service in a MRSN, the requester agent would grant more trust to a family member rather than a friend, still less to a colleague to look after his child. Such trust consideration influences the agent's decision making when choosing a counterpart to interact with. In this case, trust should be added as an additional step after traditional phases in services discovery. So that once a requester agent has discovered which providers perform the requested service, trust may be useful to rank them in terms of trustworthiness for that service. In this work, we take the stance that trust is compositional, the overall evaluation on an agent is obtained as a result of the combination of different measures. Concretely, trustworthiness towards providers and their offered services is built upon two dimensions: trust in sociability and trust in expertise. High sociability gives an indication about service provider confidence and high expertise gives rise to good services. More specifically, trust in sociability is computed to evaluate the level of trust a requester may have in a provider. Based on the analysis of the MRSN graph and the extracted information, three measures are computed: the centrality degree of a provider which gives an indication about it social power (viz. social position), the proximity in the SN between the provider and the requester (viz. social proximity), and the ties between them based on the comparison of their profiles and social acquaintances (viz. social similarity). All of theses measures take into account the semantic aspect of the relationships between agents. Trust in expertise is computed to evaluate the level of trust a requester may have in a service based on the number and ratio of successful past use of the service (viz. usability and reliability) as well as requester's feedbacks (viz. quality rate).

Second, for a given query, locating trustworthy providers with good services is often beyond the local view of the requester agent. It requires the involvement of its social acquaintances (i.e., neighbors in the MRSN graph) and even includes the involvement of the acquaintances' acquaintances. Agents can help each others to find good services by recommending trustworthy providers that have been useful to them in the past. Furthermore, the distributed nature of the environment and the network topology contains other aspects that need to be taken into account. To propagate the search in SN, we rely on a *referral system*. A referral system is a multi-agent system (MAS) in which agents represent users and act on behalf of them to discover trustworthy providers with good service. Endowed with a limited view of the environment, agents cooperate by giving, pursuing and evaluating referrals [YS03]. In both [YS03] and [YVS99], the referral system is used for searching SN but these approaches present the following limits: they assume a centralized decision-making wherein the requester's agent collects all possible referrals and decides to continue the search by contacting some of the suggested referrals.

In this work, we distribute the searching process by spreading out the query among cooperative agents with respect to the SN topology. Unlike approaches in [YS03, YVS99], we do not consider a fixed central agent. Conversely, agents cooperate to locate trustworthy recommenders ¹ and providers with good services based on a distributed knowledge about their acquaintances' expertise. Cooperation is established with a decentralized decision-making. The remainder of the paper is organized as follows. In the next section we present related work on service discovery with trust, and on agent-based models in distributed environments. Section 3 defines the main concepts used in this work and presents an example to illustrate our problem. In Section 4, we describe our trust model developed to evaluate the trustworthiness of providers and their offered services. Afterwards, we presents in section 5 our agent-based referral approach along with a description of our distributed search algorithm. Section 6 discuss implementation and evaluation aspects of our work. Section 7 gives the conclusion and the ongoing work.

¹Recommenders are agents that do not have useful services but may be well connected and help by proposing pertinent referrals.

2 Related Work

To propagate the search in a SN, our approach uses a referral system. A referral system is a MAS where agents are capable of giving and following referrals [YS03]. In [YVS99, YS03], a referral system is used for searching SNs with a centralized decision-making. The requester's agent collects all possible referrals and decides to continue the search by contacting some of the suggested referrals. However, due to the distributed characteristic of the SN, it is not possible to gather all information by a single agent and make thereafter a correct trust evaluation. To address that, a later work [SJ05] introduces decentralized search algorithms in networks using homophily and degree disparity. Their algorithms have been designed for searching peer-to-peer distributed networks. However, despite enhancements in decision making and searching, some SN features are missing as for example the multirelation aspect of SNs or the trust consideration between agents. Our work is similar to the work presented in [ESAA04, HYY⁺13]. Both are based on a structured peer-to-peer system. In [HYY⁺13] authors propose Chord4S, a Chord-based decentralized service discovery approach that supports service description distribution and discovery in a P2P manner. A service query can be submitted to any node, and this node, if it does not store the required service description, is able to route the query to an appropriate node for resolution. This proposal do not address trust related issues, there is no mechanism for assessing the quality of the returned services nor the trustworthiness of the providers. The discovery process is based only on the functionality of the Web services which is not sufficient in many cases where several providers may offer functionally equivalent services. To address that, Emekci et al. propose in [ESAA04] a Chord-based structured peer-to-peer framework for Web service discovery in which services are located based on their functionalities and their behavior. Once the qualifying services are discovered, they are ranked based on a reputation model. This model is defined as the combination of two measures which are the quality of the offered service and the trustworthiness in its provider. Both quality and trust measures are computed as the average of their respective ratings over last 6 months. However, this reputation model is simple and elementary. Authors did not precise how providers are evaluated nor what type of quality of service is considered. Moreover, they did not take into account the semantic aspect of the relationships between peers.

To discover trustworthy providers with good services, many approaches have investigated trust as far as expertise is concerned [LHZ⁺12, LCM12]. Trust in expertise is concerned with the evaluation of past interactions between users and services. For instance, Li et al. [LHZ⁺¹²] capitalizes on users' feedbacks including ratings, opinions and relevant comments after use to compute service reputation. Another recent work [LCM12] proposes an approach for using quality of past experiences as a criterion for web service selection, including an analysis of the different influence factors that may affect the perceived quality of the end-user. In spite of the added value of these retrieved information to evaluate trust between agents, trust in expertise dimension is insufficient to make a significant evaluation. Usually, in MAS the trust between two agents is based on another dimension called the trust in sociability. In this regard, we can find some interesting approaches that study trust from this point of view. We retain the works of Castelfranchi et al. [CF98] and Sabater et al. [SS02]. In [CF98], authors claim that the notion of trust is crucial in agent's theory and in MAS. An agent must trust another agent to delegate a task. Trust model is based on the expertise of the counter party regarding the task (viz. core trust) and on its willingness to achieve this task (viz. reliance). In [SS02], Sabater et al. propose a model for reputation called REGRET which takes into account, the social dimension. In another work [BBB10], Bansal et al. evaluate providers' trustworthiness based on the centrality degree that gives an indication on their prestige in the network. However, this is a poor definition of trust, since it just uses a single measure, the centrality degree. In this work, we evaluated trust in providers based on sociability as well as expertise. Moreover, we do not consider a fixed central agent but we promote agents to cooperate in order to locate trustworthy providers with good services based on a distributed knowledge concerning acquaintances' expertise. Cooperation is established with a decentralized decision-making. We distributed the referral system by spreading out the query among agents with respect to the SN topology.

3 Background

This section presents an example to illustrate our mechanism. This example will be used all along the paper. We describe in further detail the overall concepts used in this work, i.e., social networks, services, user needs, and software agents.

3.1 Walk-through Example

To better illustrate the service discovery issues and how they are dealt by our proposed solution, let us consider a case study on a travel scenario. Figure 1 presents a SN of a user namely Bob. Bob is modeled by agent a_0 . In this SN, there are web services from on-line travel agencies as well as freelance service providers. Providers may offer more than one service at the same time. Bob, the requester is a researcher and has to travel oversees frequently. To do this, he formulates a query containing required services as well as constraints (for example, price and total duration). The normal routine in his travel includes transportation from home to the airport, flight to the destination city, transportation from the airport to the hotel, hotel accommodation, and sight seeing the places in the available free time. Moreover, Bob may express his preferences over relationship types (for example, in figure 1 he would grant more trust to a friend members than to a partner, still less to a public one) and fixes a threshold for the trust.



Figure 1: Agent Bob's Multi relation Social Network Graph G

3.2 Concepts Definition

3.2.1 Social Networks

We consider a particular kind of complex networks, the multi-relation social network (MRSN) which takes into account the semantic aspect of the relationship linking two nodes. The relationships can be of different types. For example, if we consider two types, R_1 can be a friend relationship and R_2 can be a partner relationship. A multi-relation social network (MRSN) is modeled by an undirected graph, where nodes represent agents and, an edge between two nodes indicate a symmetric social relationship between the agents ².

Definition 1 Given a set V of agents and a set R of types of symmetric relationships with $R = \{R_1, R_2, ..., R_r\}$, a multi-relation social network (MRSN) is a connected undirected graph G = (V, E), in which each $E_i \subset E = \{E_1, E_2, ..., E_r\}$ is the set of edges with respect to the *i*-th relationship. In other words, an edge $(a_k, a_j) \in E_i$ represents a social relationship of type R_i between a_k and a_j .

In the MRSN graph, the notion of neighborhood of an agent can be expressed as follows:

Definition 2 Given an MRSN graph G = (V, E), the neighborhood of an agent a_k regarding a type of relationship $R_i \in \{R_1, R_2, ..., R_r\}$, denoted $N_{R_i}(a_k)$, is defined as $N_{R_i}(a_k) = \{a_j \in V \mid (a_k, a_j) \in E_i\}$.

For example, the neighborhood of agent Bob in figure 1 according to friend relationship is $N_{friend}(a_0) = \{a_3, a_6\}$. In the MRSN, each agent a_k cooperate with a subset of agents, called SA_k , that represents a_k 's local view in the MRSN such as $SA_k = \bigcup_{R_i} N_{R_i}(a_k)$. For instance, in figure 1, $SA_0 = \{a_2, a_3, a_4, a_5, a_6, a_7\}$ is the social acquaintance of agent a_0 .

3.2.2 Services

A service is described in terms of functionality, inputs and outputs.

Definition 3 A service s is a triplet (in, out, f) where in is a set of inputs required to use the service, out is a set of outputs provided at the completion of the service and f is a functionality describing the provided capacity.

For example, consider a flight service offered by the agent a_4 in figure 1. This service is described as follow: $in = \{departure, arrival, check in, check out\}, out = \{online invoice, price\}$ and, $f = \{Flight Booking\}$.

 $^{^{2}}$ We make the hypothesis that there is only one type of relationship between two nodes.

3.2.3 User needs

A user communicates his needs by expressing a set of requirements and constraints on the requested services.

Definition 4 A query Q is a quadruple (F, C, U, α) where $F = \{s_1, s_2, ..., s_l\}$ are required services, $C = \{c_1, c_2, ..., c_r\}$ is a set of constraints given over services, $U : R \to \mathbb{N}$ is an utility function expressing user's preferences over relationship types, and $\alpha \in [0, 1]$ is a trust threshold.

To illustrate the user query, let us consider the previous example. Suppose that Bob, the requester, has the intention to leave from Berlin to Paris to attend a conference on the May 5th until May 11th. He requires a 3-star hotel at least which is max 20 mn walking distance from the conference venue. Bob prefers to travel with friends rather than with partners still less by public transports and he requires service providers with a trust value greater than 0.7. In this example, the user query is defined by $F = \{Transportation service, Hotel service\}$, and $C = \{Hotel \geq 3\text{-star}, Hotel location: "max 20 mn walking distance"}\}$. His preferences over relationship types are expressed as follows: $R_{friend} \succ R_{partner} \succ R_{public}$. These preferences are modeled in the query by the utility function such as: $U(friend) = a, U(partner) = b, \text{ and } U(public) = c, \text{ with } a \geq b \geq c$. For example, if the requester prefers twice more the first relationship to the second one, than the utility of the second relationship will be double the value of the first one (i.e. b = 2a). This means that the utility of friend type edge is twice the utility of the partner type edge. This allows to favor paths that uses the minimum number of different kind of relationships regarding the requester's preferences. If we suppose that a = 1, Bob's preferences become U(friend) = 1, U(partner) = 2 and U(public) = 4, and his trust threshold $\alpha = 0.7$.

3.2.4 Software Agents

In the MRSN graph, each user is assigned an autonomous agent a_k endowed with a bounded set of services with their advertised QoS values and acts on behalf of him to discover trustworthy providers with good services.

Definition 5 An agent a_k is defined as a 3-modules structure $\langle \mathcal{RM}, \mathcal{CM}, \mathcal{KM} \rangle$ with:

- $\mathcal{RM} = \langle MC, TC \rangle$ is the reasoning module with MC the matching component and, TC the trust component.
- CM is the communication module whose role is to structure the messages built by the agent and handles the received ones.
- $\mathcal{KM} = \langle S_k, PIT_k \rangle$ is the knowledge module with $S_k = \{s_{k1}, \ldots, s_{km}\}$ a set of offered services and PIT_k a Personal Interaction Table. Each record in the PIT_k (see table 1) contains the following elements: an acquaintance agent $a_j \in SA_k$, the profile Pr_j of a_j , the social acquaintances set SA_j of a_j and the set of services S_j provided by a_j .

Acquaintance	Profile	Social Acquaintances	Offered Services with their QoS(Us,Re)
a_2	Pr_2	$SA_2 = \{a_0, a_1, a_3, a_5, a_8, a_{12}\}$	$S_2 = \{(s_{21}, 0.4, 0.7, 0.7); (s_{22}, 0.6, 0.8)\}$
a_3	Pr_3	$SA_3 = \{a_0, a_1, a_2, a_4, a_{11}, a_{12}\}$	$S_3 = \{(s_{31}, 0.3, 0.6, 0.7); (s_{32}, 0.7, 0.6)\}$
a_4	Pr_4	$SA_4 = \{a_0, a_1, a_3, a_7, a_8, a_{11}\}$	$S_4 = \{(s_{41}, 0.5, 0.9, 0.8); (s_{42}, 0.5, 0.9)\}$
a_5	Pr_5	$SA_5 = \{a_0, a_2, a_8, a_9\}$	$S_5 = \{(s_{51}, 0.4, 0.9, 0.9); (s_{52}, 0.6, 0.9)\}$
a_6	Pr_6	$SA_6 = \{a_0, a_8\}$	$S_6 = \{s_{21}, 1, 0.7\}$
a ₇	Pr_7	$SA_7 = \{a_0, a_4, a_8, a_{10}\}$	$S_7 = \{(s_{71}, 0.4, 0.8, 0.7); (s_{72}, 0.6, 0.8)\}$

Table 1: Agent a_0 Personal Interaction Table (PIT₀)

The values in the table are displayed by a get() method and a put() method that automatically replaces any preexisting value that is associated with the specified key with the new value (see Algorithm 1 in section 5.3).

The knowledge module \mathcal{KM} provides data structures where a_k stores information concerning both himself (the profile of its associated user Pr_k and the set of offered services with their advertised QoS values S_k) and its acquaintances in the SA_k . Through its reasoning module \mathcal{RM} , a_k can evaluate trust of its acquaintances before interacting with them and can also check whether him or an acquaintance of its SA_k is able to provide suitable services. MC is initialized with a set of methods that can be undertaken to accomplish a pattern matching between agent's offered services and needed functionalities. To interact with other agents, a_k uses its communication module \mathcal{CM} .

We remind agent's characteristics:

- Agents have limited view in the SN, they know only agents that belong to their social acquaintance set.
- Agents perform decentralized decision-making for contacting other agents.
- Agents do not act maliciously and don't lie while exchanging message, they have the inhered good will to fulfill user's needs.

4 Trust Computation

Trustworthiness is evaluated over two dimensions: sociability and expertise. High sociability indicates relevant providers and good recommenders, and high expertise gives rise to good services.

4.1 Trust in Sociability

Trust in sociability (ST) evaluates the social trustworthiness that an agent a_k may have towards an agent a_j . Based on the analysis of the MRSN graph and the extracted information, three measures are computed (see [LEHP12] for more details).

- Social Position Measure (SPo). The social position of an agent a_j is computed based on its centrality degree in order to give an indication about its social power (see formula in table 2).
- Social Proximity Measure (SPr). It is a distance metric between a pair of agents, defined by the average cost of their path (see formula in table 2).
- Social Similarity Measure (SSi). The social similarity between two agents is computed based on their profiles and the MRSN graph. $SSi(a_k, a_j)$ is an aggregation of two measures:
 - Profile Similarity (PS). In SNs, a profile consists of a set of items structured into a set of fields, each field containing one or several values (e.g. gender=[female], music-likes=[folk, jazz, pop]). In the former, the field is called a single-valued field and in the latter a multi-valued field. The aim of PS is to compare values of fields in requester profile to those in provider profile in order to determine how much requester and provider are similar (see formula in table 2).
 - Neighborhood Similarity Measure (NS). The neighborhood similarity between two agents is computed based on the comparison of their social acquaintances within the MRSN graph (see formula in table 2).

We define the overall measure of social similarity, $SSi(a_k, a_j)$ as the product of the two above measures: $SSi(a_k, a_j) = NS(a_k, a_j) \times PS(a_k, a_j).$

After computing these measures, a vector M_j associated with each agent a_j is defined as $M_j = (SPo(a_k), SPr(a_k, a_j), SSi(a_k, a_j))$. Considering the vectors of all acquaintances, a matrix $M = (M_{jt}, a_j \in SA_k \text{ and } 1 \leq t \leq 3)$ is built. $ST(a_k, a_j)$, is then computed using a Simple Additive Weighting technique:

$$ST(a_k, a_j) = \sum_{t=1}^3 \lambda_t \cdot M'_{jt}(a_k, a_j)$$

where $\lambda_t \in [0, 1]$ and $\sum_{t=1}^{3} \lambda_t = 1$. λ_t represents the weight of the *t*th social measure; and $M' = (M'_{jt}, a_j \in SA_k \text{ and } 1 \leq t \leq 3)$ is the matrix of vectors obtained after the scaling phase which transforms every measure value of M_{jt} vector into a value M'_{jt} between 0 and 1.

4.2 Trust in Expertise

A good agent should not only be socially trustworthy but also sufficiently expert. Inspired from [LCM12], we define trust in expertise ET of an agent a_j based on three attributes:

• Usability: is the percentage of successful use of an agent's service s_{jl} compared to the other services it offers: $Us(s_{jl}) = \frac{Nb_{success}(s_{jl})}{\sum_{l=1}^{m} Nb_{success}(s_{jl})}$ where $Nb_{success}(s_{jl})$ is the number of successful executions of s_{jl} . This means that the more the service s_{jl} is sought for in the social network the more a_j is recognized as an expert in this field.

Table 2: Trust in sociability measures			
Trust measure	Formulation		
SPo	$SPo(a_j) = \sum_{i=1, \forall a_l \in SA_j}^{ R } w_i \cdot b^i(a_j, a_l)$ where w_i is the weight of		
	the relation R_i computed as $\frac{1}{U(R_i)}$; and $b^i(a_j, a_l) = 1$ iff a_j and		
	a_l are directly connected with an edge of a relation type R_i , 0		
	otherwise.		
SPr	$SPr(a_k, a_j) = \frac{\sum_{l=1}^{d} U((a_l, a_{l+1}))}{d}$ where $U((a_l, a_{l+1}))$ is the cost of		
	the edge $(a_l, a_{l+1}) \in path$ given by the utility function U regarding		
	the requester agent preferences.		
PS	$PS(a_k, a_j) = \frac{1}{ I } \times \sum_{i \in I} \beta_i \cdot S_i(a_k, a_j)$ where $S_i(a_k, a_j)$ is the		
	similarity between the <i>i</i> th items of a_k and a_j using Burnaby mea-		
	sure [Bur70], I is the set of items in profiles and β_i is a weight		
	attributed to the item i reflecting its importance in the profile		
	description.		
NS	$NS(a_k, a_j) = \sum_{i=1}^{ R } w_i \cdot \delta^i(a_k, a_j)$ with $\delta^i(a_k, a_j) = \frac{1}{1 + jac^i}$		
	where $w_i = \frac{1}{U(R_i)}$ and $jac^i = \frac{y_i + z_i}{x_i + y_i + z_i}$ is the Jaccard distance		
	between a_k and a_j according to the relationship R_i such as $x_i =$		
	$ N_{R_i}(a_k) \cap N_{R_i}(a_j) , y_i = N_{R_i}(a_k) - x_i, z_i = N_{R_i}(a_j) - x_i.$		

- Reliability: is the probability that a service s_{jl} is operational at the time of invocation: $Re(s_{jl}) = \frac{Nb_{success}(s_{jl})}{Nb_{invoc}(s_{jl})}$ where Nb_{invoc} is the number of invocations of s_{jl} .
- Quality rating: is the quality of a service s_{jl} . Once a service s_{jl} is successfully used by an agent a_k , an evaluation $\nu \in [0, 1]$ is attributed by the user to this service. We designate by $Eval_x(a_k, s_{jl})$ the average of the quality rating of s_{jl} for x uses.

$$Eval_x(a_k, s_{jl}) = \begin{cases} 1 & \text{if } x = 0\\ \frac{Eval_{x-1}(a_k, s_{jl}) * (x-1) + \nu}{x} & \text{otherwise} \end{cases}$$

Trust in expertise $ET(a_k, a_j)$ measures, the ability of a_j to meet the expectations of a_k . It's an overall score established on the basis of the three aforementioned attributes and over the *m* services offered by a_j .

$$ET(a_k, a_j) = \frac{\sum_{l=1}^m (Us(s_{jl}) \times Re(s_{jl}) \times Eval_x(a_k, s_{jl}))}{m}$$

4.3 Global Trust

The global trust $Trust(a_k, a_j)$ that an agent a_k has for an agent a_j is a weighted sum that depends on the expertise as well as on the sociability of a_j .

$$Trust(a_k, a_j) = w \times ST(a_k, a_j) + (1 - w) \times ET(a_k, a_j)$$

5 Distributed Trust-based Service Discovery

In this paper, our goal is to search user's required services through a distributed trust-based service discovery. In the following, we present our original search mechanism based on a distributed referral approach.

5.1 Distributed Referral Approach

In [YS03] and [YVS99], a referral approach is used with a centralized decision-making : when an agent receives a request for expertise either it tries to match it against its own functionalities, or generates a list of possible referrals using its user contact file, i.e., social acquaintances. The query originator's agent collects all possible referrals, and decides to continue or not the search by contacting some of the suggested referrals. In this work, we propose a distributed version of the referring process where, each agent acts autonomously and decides locally to continue or not the search by propagating the query in its social acquaintance without coming back to the requester agent. Some agents may be good providers and some others may be bad providers but may be well connected and thus, may recommend good referrals. In our approach, we consider three different roles that an agent may take: *requester*, *provider/referral* or, *recommender*.

5.2 Hypotheses

Before describing the algorithm, let us give some hypotheses. Recall that each agent a_k offers a set of services $S_k = \{s_{k1}, \ldots, s_{km}\}$ and has a set of social acquaintances SA_k . Additionally, each agent a_k encompasses the following variables:

- A status variable, denoted $status_k$, that represents the role taken by an agent during the search process. It takes its value in the set $\{\perp, Req, Prov, Rec, Stop\}$ with Req for requester, Prov for provider, Rec for recommender, and Stop for stopper. Initially, the status of all agents equals \perp except for the requester agent a_r whose status is equal to Req.
- A distance variable, denoted $dist_k$, that represents the distance of an agent to the requester agent a_r . Initially, this variable equals 0 for the requester agent a_r and $+\infty$ for all other agents.
- A counter variable, denoted $count_k$, that serves to count the number of messages received from its acquaintances. Initially, this variable equals 0 for all agents.
- A set of trustworthy agents, denoted LTA_k , representing the set of potential providers belonging to the social acquaintances SA_k of a_k . Initially, LTA_k is an empty set for all agents.
- A set of trustworthy agents, denoted LRA_k , representing the set of potential referrals belonging to the social acquaintances SA_k of a_k . Initially LRA_k is an empty set for all agents.

The message structure is adapted form ACL syntax: *<performative, receiver, content>*.

5.3 Distributed Trust-Based Search Algorithm

The distributed search is presented in Algorithm 1. It ensures the propagation of search via navigation in the SN. The inputs are G a MRSN graph and $Q = (F, C, U, \alpha)$ (see Def 4) a query transmitted by a requester to its agent. As the output of the algorithm, each agent knows its status regarding the query (provider, recommender or stopper) and its distance to the requester agent in the provider-recommender chain. A provider-recommender chain is a sequence of agents in G starting from the requester agent a_r and leading to a provider in which all agents are either providers or recommenders.

The algorithm is composed of four procedures and begins with an initiator, the requester agent a_r , acquiring the requester role (*status_r* = *Req*), setting its distance to itself to 0, and starting the discovery process by sending an REQUEST message to all its acquaintances (see lines 2 to 5). After that, a_r waits ³ for an INFORM message from each of its acquaintances (see line 6). Once a_r received all REQUEST messages (one message per acquaintance), it determines using its *TC*, a subset of trustworthy agents *LTA_k* from the set of its *SA_k* (see line 8) and sends them a PROPAGATE message (see lines 10 to 13).

The behavior of an agent a_k is event-driven: the reception of a message triggers the execution of a procedure depending on the received message. An agent a_k that receives an REQUEST message, sends an INFORM message containing information about its profile Pr_k , offered services S_k with their advertised QoS values and its social acquaintance SA_k (see line 21 to 23). An agent a_k that receives an INFORM message from an acquaintance a_j , increments its message counter and upgrades a_j 's record in PIT_k as means to upkeep its beliefs regarding a_j 's current offered services with their QoS values, profile and, its social acquaintance SA_i (see line 24 to 27).

An agent a_k that receives a PROPAGATE message, compares the received distance value with its current distance value. If the received value is less than its current distance value, then a_k sets $dist_k$ to the new value which represents its distance in the provider-recommender chain to the requester agent a_r (see lines 29 to 30). After that, a_k sends an REQUEST message to all its acquaintances and waits for an INFORM message from each of its acquaintances (see lines 31 to 34).

 $^{^{3}}$ As agents have the good will to help, we suppose that all acquaintances answer. If not, we consider that there is a failure in the system which is out of scope of this paper.

Algorithm 1 Distributed Trust-based Search Algorithm

variables: $status_k \in \{\bot, Req, Pro, Rec, Stop\}, dist_k \in \{0, \ldots, n\}, count_k \in \{0, \ldots, |SA_k|\}, LTA_k \text{ a set of trustworthy potential providers, <math>LRA_k$ a set of trustworthy potential referrals.

1: procedure PROPAGATION (F, C, U, α) if $(status_k == Req)$ then 2: for all $a_i \in SA_k$ do 3: \triangleright SA_k set of social acquaintance agents 4: $send(Request, a_i);$ ▷ The REQUEST message end for 5:wait $(count_k == |SA_k|);$ 6: $count_k \leftarrow 0;$ 7: $LTA_k \leftarrow \{a_j \in SA_k \mid Trust(a_k, a_j) > f_\alpha(dist_k)\};$ 8: 9: end if if $(status_k == Req)$ or $(status_k == Prov)$ then 10: for all $a_i \in LTA_k$ do 11:send(*Propagate*, a_i , F, U, α , $dist_k$); 12:▷ The Propagate message 13: end for end if 14:if $(status_k = Rec)$ then 15:for all $a_i \in LRA_k$ do 16:send(*Propagate*, a_i , F, U, α , $dist_k$); 17:▷ The Propagate message end for 18: end if 19:20: end procedure 21: **procedure** RECEIVE(*Request*, a_j) 22:send(Inform, a_i , Pr_k , S_k , SA_k); ▷ The INFORM message 23: end procedure 24: procedure RECEIVE(Inform, a_i , Pr, S, SA) 25: $count_k \leftarrow count_k + 1;$ 26: $PIT_k.put(a_i, Pr, SA, S);$ 27: end procedure 28: **procedure** RECEIVE(*Propagate*, a_i , F, C, U, α , dist) if $(dist_k > dist + 1)$ then 29: $dist_k \leftarrow dist + 1;$ 30: for all $a_l \in SA_k$ do 31: $send(Request, a_l);$ 32: end for 33: wait $(count_k == |SA_k|);$ 34: 35: $count_k \leftarrow 0;$ if $((\exists s_{kl} \in S_k / s_{kl} \equiv s \in F) \land (s_{kl} \models C))$ then 36: $LTA_k \leftarrow \{a_j \in SA_k \mid Trust(a_k, a_j) > f_\alpha(dist_k)\};$ 37: $status_k \leftarrow Pro;$ 38: PROPAGATION (F, C, U, α) ; 39: ▷ In order to find other potential providers in G else 40: $LRA_k \leftarrow \{a_j \in SA_k \mid Trust(a_k, a_j) > f_\alpha(dist_k) \text{ and } \exists s' \in PIT_k.get(a_j), | s' \equiv s \subset F \land (s_{kl} \models C\};$ 41: if $(LRA_k \neq \emptyset)$ then 42:43: $status_k \leftarrow Rec;$ PROPAGATION(F, C, U, α); 44: ▷ In order to find other potential providers in G else 45: $status_k \leftarrow Stop;$ 46: end if 47:end if 48: end if 49: 50: end procedure

Once an agent a_k received all INFORM messages (one message per acquaintance), it checks its set of services S_k and based on its MC, it verifies if one of its services fits user's needs F given its constraints C (see line 36). In both cases, a_k computes its trust threshold and trust value towards all its acquaintances using its TC. The computation of a_k trustworthiness towards an acquaintance a_j , $Trust(a_k, a_j)$ is an aggregated value over a_j 's expertise and sociability (see Section 4). The computation of a_k 's trust threshold is done based on the following observation: if we consider that the agent requester a_r is a point of reference in the graph, the longer the provider-recommender chain is, the less trustworthy the contacted agents are. Therefore, we have to adapt the value of trust threshold (α) initially fixed by a_r to the current search depth (current length of the provider-recommender chain). For this, we define a function denoted $f_{\alpha}(dist_k)$ which adjusts the threshold value of a_k depending on its distance $dist_k$ to a_r in the provider-recommender chain as follows: $f_{\alpha}(dist_k) = (1 - \alpha) \times (1 - e^{\frac{-dist_k}{D}}) + \alpha$, where D is the graph diameter.

Once a_k has computed its trustworthiness towards all its acquaintances and its trust threshold, it determines a subset of trustworthy agents. There are two possible cases, the first is that a_k has a service $s_{kl} \in S_k$ that offers the same capacity as a requested service $s \in F$, then a_k determines the LTA_k subset. To be in this subset, an acquaintance a_j has to have a trust value $Trust(a_k, a_j)$ greater than $f_\alpha(dist_k)$ (see line 37). Thereafter, a_k gets the provider role (status_k = Pro) and executes the PROPAGATION procedure (see lines 38 to 39) that sends a PROPAGATE message to all acquaintances in LTA_k (see lines 10 to 12). In a second case, a_k does not provide the desired service, it determines the subset of agents LRA_k which are not only trustworthy $(Trust(a_k, a_j) > f_\alpha(dist_k))$ but also that offers a desired service $(\exists s' \in PIT_k.get(a_j), / s' \equiv s \in F)$ (see line 41). If a_k has no potential referrals (i.e. either they are not enough trusty to be recommended or they do not offer at least one requested service), then a_k stops the query propagation and takes the stopper role (status_k = Stop) (see line 45 to 46). Otherwise, a_k gets the recommender role (status_k = Rec) and executes the PROPAGATION procedure (see lines 42 to 44) that sends a PROPAGATE message to all acquaintances in LRA_k (see lines 15 to 18). This way of progressive spreading is the original mechanism of referral systems in which the propagation of the search is ensured via navigation in the graph. In this case, although a_k does not provide any desired service, it can participate in the discovery process while leading to relevant agents.

Let us consider our travel scenario in which we apply the algorithm. Recall that required services are $F = \{ Transportation service, Hotel service \}, a_0$ who is the requester agent, launches the discovery process in the SN graph by contacting trustworthy agents of its SA_0 (a_2 and a_7 are not trustworthy). An agent who is able to participate to the query achievement is considered as a provider (a_4 and a_5) whereas, all intermediate agents leading to providers are called recommenders. In this example, a_3 and a_6 are recommenders since they do not have useful services but they are well connected and lead to relevant providers (for example, a_3 recommends a_1). As described above, each contacted agent propagates the query within its social acquaintance using the same mechanism and only agents which are sufficiently trustworthy and with useful service(s) are detected.



Figure 2: Service providers' discovery

At the end of this search process we obtain an annotated graph as shown in figure 2 wherein providers are recognized (checked nodes). Note that some agents which have services to respond to the user query, a_{10} and a_{12} , are not discovered since they are not enough trusty to be recommended.



Figure 3: Percentage of discovered providers as a function of: (a) trust threshold (b) the length of providerrecommender chain.

6 Experimental Results

We have developed a prototype using Java 1.7 and the Jade⁴ multi-agent platform. The MRSN graph data was stored in a GML format⁵. All experiments were run on a 3.1GHz Core(TM) i5-2400 running windows 7, with a 8Go of RAM. Simulations were done over 11 MRSN graph randomly generated. The number of agents varies from 500 to 10000. Each graph contains 3 types of relationship and the requester preferences over relationships types are equal to U(R1) = 1, U(R2) = 2, U(R3) = 4. We run each simulation with four initial values of α , i.e., $\alpha = 0, 0.2, 0.4, 0.6$. We average the obtained results over 5 instances of each graph. Services were also randomly distributed among agents thus, each agent in the graph may be a potential provider depending on the requested service. For sake of simplicity, we suppose for all experiments that the requester agent needs only one service *s* at a time. We evaluate our algorithm by computing the percentage of discovered providers (see Fig 3 a and b).

6.1 Percentage of discovered providers as a function of trust threshold

We investigate the impact of trust threshold on the discovery process (Fig 3.a). We compute the percentage of discovered providers while varying trust threshold as depicted in figure 3.a. For $\alpha = 0$, almost all providers are discovered. This is equivalent to an exhaustive search that takes into account only the functional aspect of the provider and ignores its social aspect. As expected, the number of discovered providers decreases while the threshold increases. For high values of α , agents explore less search space in the MRSN which reduces the number of discovered providers.

6.2 Dispersion of discovered providers as a function of the length of provider-recommender chain

We study how the number of discovered providers is scattered in the MRSN in terms of provider-recommender chain length for a given requester agent query (Fig 3.b). Using the trust measure, we compute the percentage of discovered providers (averaged over all graph instances) for different ranges of length. Now, let us look at figure 3.b; this figure displays the providers' dispersion for different trust threshold. For each threshold value, the average number of discovered providers tends to level off once the length reach a certain point. There is a trade-off between the chain length and the number of discovered providers: there is seven times more chances to find a trustworthy provider in a chain length equal to 7 than in chain length equal to 3 (from 14% to 80%). We noticed also that for values of α greater than 0.4, trustworthy providers are discovered relatively far away from the requester. This indicates that many of discovered providers are weakly connected to the service requester and require a long chain of recommenders to be reached. These results help us to fix the length of the providerrecommender chain as a parameter. Setting a maximum length to 6 gives us the possibility to control the scope of search which would substantially limit the computation costs.

7 Conclusion and Future Work

In this work, we have presented a distributed approach for trust-based service discovery to satisfy user's needs in social networks. We propose to evaluate trustworthiness towards service providers and their services over two dimensions namely sociability and expertise. The discovery process is done by a distributed referral system that spreads out the query among agents with respect to the SN topology. As future work, we plan to compare our service discovery approach with a non trust-based one. By means of simulation, we plan to evaluate the impact

⁴Telecom Italia Lab. JADE 4.3 http://jade.tilab.com/.

 $^{{}^{5}\}text{Graph Modeling Language, 1997, http://www.fim.uni-passau.de/en/fim/faculty/chairs/theoretische-informatik/projects.html.}$

of our social trust at agent-decision making during interaction and thereafter the quality of the underlying service discovery. We also would like to investigate the composition of provider-recommender chain by keeping within each service provider the identity of its acquaintances that recommend them. Finally, we intend to explore the extension of our model to perform a service composition built upon a coalition formation of trust-based discovered services. The idea is to achieve complex user's needs through a service composition where discovered agents communicate and coordinate within an organizational structure.

References

- [BBB10] S. K. Bansal, A. Bansal, and M. B. Blake. Trust-based dynamic web service composition using social network analysis. In *BASNA*, pages 1–8, 2010.
- [Bur70] T. Burnaby. On a method for character weighting a similarity coefficient, employing the concept of information. *Mathematical Geology*, 2:25–38, 1970.
- [CF98] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *ICMAS*, pages 72–79. IEEE Computer Society, 1998.
- [ESAA04] F. Emeki, O. D. Sahin, D. Agrawal, and A. El Abbadi. A peer-to-peer framework for web service discovery with ranking. In *ICWS*, pages 192–199. IEEE Computer Society, 2004.
- [GA05] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In Proceedings of the 2005 ACM workshop on Privacy in the electronic society, WPES '05, pages 71–80. ACM, 2005.
- [HYY⁺13] Qiang H., Jun Y., Yun Y., Ryszard K., and Hai J. A decentralized service discovery approach on peer-to-peer networks. *IEEE Transactions on Services Computing*, 6(1):64–75, 2013.
- [LCM12] F. Lalanne, A. Cavalli, and S. Maag. Quality of experience as a selection criterion for web services. In Signal Image Technology and Internet Based Systems, Eighth International Conference on, pages 519–526, 2012.
- [LEHP12] A. Louati, J. El Haddad, and S. Pinson. Trust-based service discovery in multi-relation social networks. In *ICSOC*, pages 664–671. Springer-Verlag, 2012.
- [LHZ⁺12] M. Li, Z. Hua, J. Zhao, Y. Zou, and B. Xie. Arima model-based web services trustworthiness evaluation and prediction. In *ICSOC*, pages 648–655. Springer-Verlag, 2012.
- [MHDC10] A. Maaradji, H. Hakim, J. Daigremont, and N. Crespi. Towards a social network based approach for services composition. *IEEE International Conference on Communications*, pages 1–5, 2010.
- [MSW⁺11] Z. Maamar, P. Santos, L. Wives, Y. Badr, N. Faci, and J.P.M. de Oliveira. Using social networks for web services discovery. *IEEE Internet Computing*, 15:48–54, 2011.
- [PKPS02] M. Paolucci, T. Kawamura, T.R. Payne, and K.P. Sycara. Semantic matching of web services capabilities. In Proceedings of the First International Semantic Web Conference on The Semantic Web, pages 333–347. Springer-Verlag, 2002.
- [RKM04] J. Rao, P. Küngas, and M. Matskin. Logic-based web services composition: From service description to process model. In *ICWS*, pages 446–453. IEEE Computer Society, 2004.
- [SJ05] O. Simsek and D. Jensen. Decentralized search in networks using homophily and degree disparity. In IJCAI, pages 304–310. Morgan Kaufmann Publishers Inc., 2005.
- [SPAS03] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantics*, 1:27–46, 2003.
- [SS02] J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In C. Castelfranchi and L. Johnson, editors, *AAMAS*, pages 475–482. ACM Press, 2002.
- [YS03] B. Yu and M. P. Singh. Searching social networks. In AAMAS, pages 65–72. ACM, 2003.
- [YVS99] B. Yu, M. Venkatraman, and M. P. Singh. A multiagent referral system for expertise location. In In working notes of the AAAI workshop on intelligent information systems, pages 66–69, 1999.