Similarity Flooding for Efficient Distributed Discovery of OWL-S Process Model in P2P Networks

Adel Boukhadra*, Karima Benatchba, Amar Balla

Ecole national Supérieur d’Informatique, PB 68M, Oued-Smar, Alger, Algérie

Abstract

Due the increasing number of published Semantic Web services (SWs) rendered the distributed discovery within repositories a critical issue and a major problem that can reduce the capability and functionality of SWs in terms of efficiency and scalability. Peer-to-Peer (P2P) computing is considered as the most dominant technology to discover new distributed and heterogeneous collaborative applications for SWs. In this paper, we propose an efficient approach for improving the performance and effectiveness of automatic and cooperative discovery of large-scale distributed systems in the unstructured P2P networks. The approach exploits a scalable epidemic algorithm that uses different sources of network knowledge, such as exponential distribution, to fulfill the users requirements in order to ensure high recall, further reduce the number of messages exchanged and reduce the execution time for discovering SWs in the unstructured P2P network. In order to improve the applicability of the scalable epidemic algorithm for discovering SWs, we propose the semantic matching of OWL-S process model which improves the recall while keeping an acceptable matching quality level. The experimental results show that our efficient approach is able dynamically to adapt to network changes and preserve high levels of recall.

Keywords: SWs; Distributed Discovery; P2P Computing; Semantic Matching of OWL-S process model; Similarity Measures.

1. Introduction

Current trends in service discovery are essentially based on centralized discovery methods, where Web services are described by service interface functions (WSDL) and they publish their capabilities and functionalities with a directory (such as UDDI, ebXML)\textsuperscript{1,2,12}. These directories are restricted by the syntactic description of the functionality of service as they are known for their low accuracy and poor performance, and sometimes for their low availability. Centralized discovery methods cannot manage large and continuously growing spaces of Web services with reasonable resolution times. Moreover, the centralized methods of published services suffers from problems such as high operational and maintenance cost. The centralized methods do not ensure the required scalability to support the dynamic, flexible and evolutionary environment\textsuperscript{5,10,11}.

* Adel Boukhadra. Tel.: +213777697100
E-mail address: a_boukhadra@esi.dz
Discovering of SWs in a distributed registry is becoming a real challenge and an important task in the area of Services Oriented Computing (SOC) due to availability of various SWs which provide similar functional requirements over the network. It makes an ability to locate capabilities and components in a distributed registry in a large-scale, contributing to the resolution and simplification of a very complicated problem expressed by the user. It allows, in other words, to fulfill the users requirements evolve over time. Its not easy for the users to effectively discover and share these Web services which satisfies his specific requirements as the number of retrieved services is huge.

We argue that the service discovery should take the OWL-S process model to find all the Web services in the right order. The idea behind is to purpose matching technique that enables semantic interoperability. The purpose of matching technique is a process to discover the correspondences between entities in different OWL-S process model. These relationships are discovered through the similarity measure between those entities in OWL-S process model. There is great interest in the field of heterogeneous knowledge management for ontology matching. This technology offers great potential to overcome problems of heterogeneous and distributed information over the network.

More importantly, our main work is following: (1) we define a similarity flooding to reduce considerably the execution time and message overhead on large-size P2P networks for disseminating request among the peers by using the exponential distribution and (2) we propose an epidemic discovery algorithm based on semantic matching technique of OWL-S process model to solve and perform the discovery of SWs in an automatic and flexible manner. The goal in our work is to create a collaborative workspace which each peer can exploit the experiment of other peers by sharing their SWs for contributing to resolve discovery queries raised by these peers, without having any knowledge about each other.

The paper is organized as follows. Section 2 presents the related work on distributed discovery of SWs in P2P networks. Section 3 details our P2P approach for SWs discovery. An experimental evaluation is presented in Section 4. Section 5 concludes the paper and gives our future research focus.

2. Related Work

Recently, there is much work related to service discovery on P2P computing. Several proposals have emerged for future enhancements to the distributed discovery of SWs in P2P computing.

In 2, present a technique to improve discovery and composition in unstructured P2P service networks, based on a probabilistic forwarding algorithm driven by network knowledge, such as network density, and traces of already discovered service compositions (CONs). The technique aims at reducing the composition time and the messages exchanged during composition, relying on two considerations: if the network is dense, forwarding can be limited to a small number of neighbors; if the network is semi-structured in CONs, forwarding can be directed to the super peers that may own the desired information. The approach improves the discovery and composition process by using distributed bidirectional search. The benefit is twofold: first, it is possible to have concurrent searches in a P2P service network in both goal directions (from pre- to post- and from post to preconditions), reducing the response time when solutions are present; second, when no complete solution for a goal is present, gaps in partial found solutions can be identified. This way, it is possible to have feedbacks about users’ most required unavailable business operations, allowing providers to discover new business opportunities.

In 10, authors propose the P2P-based Semantic Driven Service Discovery (P2P-SDSD) framework to enable cooperation and communication based on a semantic overlay that organizes semantically the P2P-integrated knowledge space and emerges from local interactions between peers. The semantic overlay can be seen as a continuously evolving conceptual map across collaborative peers that provide similar services and constitute synergic service centers in a given domain. The semantic overlay enables effective similarity based service search and optimization strategies are defined for request propagation over the unstructured P2P network keeping low the generated network overload. Each collaborative peer in the unstructured P2P network has a local knowledge infrastructure constituted by: (i) UDDI Registry; (ii) Peer Ontological Knowledge, that provides a conceptualization of abstract service operations and In-put/output parameters through a given domain ontology; a conceptualization of service categories through a Service Category Taxonomy (SCT).

In 12, authors present a distributed approach to SWs publication and discovery by leveraging structured P2P network. In this work, the computers concerned constitute a P2P network to maintain the sharable domain and service ontologies to facilitate SWs discovery. When a requestor submits a semantic query for desired services, the P2P net-
work can effectively obtain semantically qualified services. The main contributions of this work can be summarized as follows: this approach introduces a semantic-based service matching rule. In order to achieve the optimal match between a query, it proposes a concept of Ordered-Concept-Tree (OCT) to semantically sort the relevant concepts for service matching. In addition, to freely share and make full use of the semantic concepts defined in ontologies for OCT construction, it also proposes a method to publish ontologies to structured P2P network. Finally, this approach presents a method for SWs publication and designs the corresponding algorithm for service discovery.

The basic idea in our work is to propose an efficient and effective approach based on P2P computing to overcome currently adopted centralized approach, which enables fully distributed and cooperative techniques for discovering SWs. The scalable approach is mapped on unstructured P2P networks by exploiting two mechanisms for improving performance and effectiveness: a similarity flooding and an epidemic discovery algorithm.

3. Component-based framework of a Peer

The key element of our efficient approach that is proposed within this paper is to propose a component-based framework of a peer, which can implement distributed discovery of SWs in the unstructured P2P network, in order to find the appropriate SWs which satisfy the user request. In particular, we expect that every peer distributing a query should be able to discover all the SWs available in the unstructured P2P network. Figure 1 depicts the overall architecture of our framework.

![Component-based framework of a Peer to discover SWs.](image)

We describe the various components of the framework, as well as all possible interactions in the following:

3.1. Scalable Epidemic Discovery Algorithm

The main idea of the epidemic discovery algorithm ensures the operation progress of the distributed discovery in the unstructured P2P network. Thus, the epidemic discovery algorithm is based on flooding among the peers for exchanging messages which are connected in unstructured P2P systems to disseminate the request in order to solve a service discovery request. In the following, we explain our epidemic algorithm (See Algorithm 1).

Each peer runs this algorithm, when it receives a user request. Initially, each peer executes the main algorithm from step 1 to step 4 where it tries to response to the request locally through local basic SWs (step 2 of algorithm 1). The procedure Discovery-Matching() in step 1, is based on a semantic matching technique in order to find appropriate SWs that fulfill the user requirements and the requested goal. To this end a hybrid matching between the request and the locally available SWs is performed ($Threshold \leq \text{Similarity}(R,S) \leq 1$).

If there is not a possibility to answer the request locally through any local basic SWs ($\text{Similarity}(R,S) < Threshold$), the peer starts a P2P discovery of SWs (step 11 of algorithm 1) to answer this request in the same peer. Moreover, if there is any basic local SWs, this peer calculates a new value of TTL$^1$ (step 7 to 9 of algorithm 1), rebroadcasts the request to its neighbors upon receiving it for the first time with a predetermined similarity $P$ (step 11); every peer has the same similarity to rebroadcast the request and executes the P2P discovery to start a collaborative workspace between different peers in the unstructured P2P network for discovering new SWs.

\footnote{TTL: Time to Live.}
Algorithm 1: Epidemic Discovery of SWs ();

Input: Upon reception of user request (R) at peer such as Search the SWs:(Input, Output, Precondition, Result and TextDescription).
Output: A set of SWs (S) which responds to the user request using matching technique between Request R and Web services S.

Begin
1 : Discovery-Matching (); // to discover a local SWs in this peer
2 : If (Threshold <= Similarity(R,S) <= 1) Then:
3 : Stop the Distributed Discovery of SWs;
4 : Send the Favorable Response; and Go to End
5 : End If;
6 : Else
7 : Calculate the TTL;
8 : If (TTL > 0) Then:
9 : TTL: = TTL - 1;
10 : Rebroadcast the request with Similarity P;
11 : P2P-Discovery (); // Send the request to all the peers in the unstructured P2P
12 : Else // networks to starts a P2P Distributed Discovery;
13 : Drop the request; and Go to End;
14 : End If; End if;
End.

After having received the request by a peer, the peer executes the request and sends the response to the transmitting peer. In addition, it decreases the TTL value by 1. If the TTL value becomes 0, destroy the message peer (step 13). If the peer forward, the message contains the new TTL value to its neighbors. By repeating this process, among all peers situated at a distance having a value less than the initial value of TTL will receive the message. Other peer in the system does not receive the message even if they have valid responses.

In this epidemic algorithm, we focus on reducing redundant query messages which unnecessarily overload the P2P system. To do so, the request is rebroadcast with a similarity $P$ which is a fixed constant determined based on extensive simulation until sufficient responses to the query are found. However, this value might not be globally optimal and so the selection of appropriate rebroadcast similarity is vital to the performance and the scalability of our scalable approach.

In this work, we propose our novel technique termed similarity flooding that will dynamically assign a rebroadcast similarity $P$ which is computed from the rebroadcast function $P = \psi(Similarity(R,S))$. The similarity flooding is simple, highly reliable and allows reaching a large number of peers in the unstructured P2P system. Although it makes better utilization of the P2P network than flooding, it consists in broadcasting the query in the unstructured P2P network till a stop condition is satisfied, which may lead to a very low query execution cost.

3.2. Similarity Flooding

The similarity flooding which is discussed in the following paragraph decreases the big overhead of network communication, while reducing the network traffic, and accelerates the discovery of SWs by providing the basic primitives for efficient query forwarding through the unstructured P2P network. To send a request that is not satisfied to relevant sources of data, the peer initiating the request sends identical messages to its neighbors. Each message contains the identifier of the sending peer (ID), a similarity measure (Similarity(R,S)) and a parameter TTL, which for example is a counter that is initialized with a positive integer. In particular, the requests are sent according to a TTL mechanism with a low TTL value to avoid network overload. Experimentation is being performed to establish the best value of TTL.
We apply results achieved by statistical analysis of random graphs in order to make any peer compute the average number of neighbors on the connectivity graph to selecting appropriate initial value for TTL of the query. Applying the results provided by 6, each peer $p$ stores a local approximation of the average number of neighbors in the network as its $state_p$; each peer $p$ performs a random selection of the neighbor $q$ to gossip with from its neighborhood as its $state_q$. This parameter represents the maximum number of hops. That is to say, the path length of each message in terms of number of peers crossed in the unstructured P2P network. The TTL between two randomly chosen peers on any unstructured P2P network is approximated as follows:

$$\text{TTL} = \frac{\ln[(N-1)(state_q - state_p) + state_p^2] - \ln(state_p)^2}{\ln(state_q/state_p)} \quad (1)$$

Where $N$ is an estimation of the total number of peers available in the unstructured P2P network. TTL between two peers presents a reasonable estimation of the distance between the originator of the query and the peer that eventually serves the requested object. TTL is actually the scope of the request: more important it is, the more there will be peers that will be visited, and the request will likely be satisfied. A large TTL also causes average response greater. Using a large TTL, this type of infrastructure can meet a maximum of elements corresponding to the search criteria. In addition, this approach is fast and reliable.

The most important factor in our work is the selection of the rebroadcast similarity $P$. A larger $P$ incurs more redundant rebroadcast while a smaller $P$ leads to lower reachability. Our motivation for this rebroadcast function is to enhance rebroadcast decision by taking into account key network parameters and peer information through TTL value and similarity measure, we can present our rebroadcast similarity function as follows:

$$\psi(\text{Threshold} < \text{Similarity}(R, S)) = 1 - \exp^{-\frac{\text{Similarity}(R, S)}{\text{TTL}}} \quad (2)$$

We opt for a rebroadcast function that uses Weibull distribution, as shown in (2), to better describe the similarity measure of peers in the unstructured P2P networks; because a high rebroadcast similarity incurs more redundant rebroadcast while a low rebroadcast similarity leads to low reachability. Moreover, peers with low values of TTL should be assigned a high rebroadcast similarity while those with high values of TTL are assigned a low rebroadcast similarity. Therefore, as the number of neighbors increases, the rebroadcast similarity should decreases. A Weibull distribution are commonly used to model lifetimes in reliability engineering due to their flexibility and versatility. The shape $\alpha$ and and scale $\lambda$ parameters can be used to describe exponential distributions when $\alpha = 1$.

We implement our matching technique for a rebroadcast function that can be used to assess the similarity function $(\text{Similarity}(R, S) \in [0, 1])$ between two concepts of OWL-S process models and to provide a matching technique between them; 0 means the concepts are totally different, 1 means that they are totally similar. Consequently, the similarity between concepts of Request $R$ and Web service $S$ is defined on the basis of their semantic relationship in the OWL-S process models.

Furthermore, in order to achieve efficient query propagation that is not satisfied in the unstructured P2P network, the peer initiating the request collaborate to propagate the query to its neighbors based on the similarity flooding dissemination mechanism. Dissemination of a query is restricted by its TTL and its similarity $P = \psi(\text{Similarity}(R, S))$.

After having received the query message by a peer, the peer applies the rebroadcast similarity function $P1 = \psi(\text{Similarity}(R, S))$ to evaluate the semantic similarity between its parameters (Input, Output, Precondition, Result and TextDescription) and those of the request, respectively.

In addition, the peer decreases the TTL value of the query by 1, and if $(\text{TTL} > 0$ and $P1 - P > \delta)$ where $\delta$ is the similarity threshold experimentally set, it executes the request and sends the response to the transmitting peer, otherwise the peer forwards the query to each of its neighbors with the similarity $P = \psi(\text{Similarity}(R, S)) \in [0, 1]$. If the $\text{TTL} = 0$ or $P1 - P \leq \delta$, destroy the message peer. In the next section we will provide our approach, evaluation results and share our experiences. The procedure (Discovery-Matching ()) made in step 1 of Algorithm 1, it used for the automatic discovery of SWs.
4. Experimental Evaluation

We conducted a series of experiments to demonstrate and evaluate the effectiveness and efficiency of our scalable approach for discovering SWs in the unstructured P2P network through event driven simulations, which are usually used to evaluate the performance of large scale P2P systems.

To evaluate and verify the performances of our efficient approach, we must run our epidemic algorithm in a large scale network and studying its impact on overall system performance. For this reason, we are using PeerSim simulator\(^7\) to simulate an unstructured P2P network and testing our approach by using a similarity flooding in the first time. To evaluate the efficiency and scalability of our approach, we must compare our approach with a simple flooding protocol Gnutella P2P protocol\(^4\). Gnutella, which is a pioneer of P2P applications, uses flooding mechanism to discover shared resources in network. The performances are devoted to measure: (1) The total number of a request submitted to a peer necessary for network operating; (2) The total number of requests produced, the number of requests forwarded, and the number of transmitted results on the network.

The data used in our experiments are OWLS-TC\(^2\) 4.0. It provides 1083 SWs written in OWL-S 1.1. It provides a set of 42 test queries which are associated with relevance sets to conduct performance evaluation experiments. With regard to the different similarity measures which are implemented in our approach, we used the Java API SIMPAC\(^3\) (Similarity Package) which represents a comprehensive library that contains all the important similarity measures. We use JWordNetSim to measure the similarity between synsets in WordNet\(^4\) 2.0.

To evaluate the quality of the epidemic discovery algorithm, we compared manually each query against all OWL-S in the unstructured P2P network. We estimated the similarity degree of each query, and also established a matching between them. We then applied the epidemic discovery algorithm to each query and we compared the matching results and the rankings that are found by the algorithm with those found manually. To do so, the following quality measures are computed:

\[
\text{Precision} = \frac{I}{N}; \quad \text{Recall} = \frac{I}{R}; \quad \text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\]  

(3)

Where, \(I\) is the set of correctly identified matching, \(N\) is the matching found by the algorithm, and \(R\) is the set of matching found manually. Usually, Precision and Recall scores are not discussed in isolation. Instead, they are combined into a single measure, such as the F-measure.

In order to evaluate the accuracy and the effectiveness of our approach, we compared the quality of the rankings found by our approach with respect to the number of the TTL and the value of \(\psi(Similarity(R, S))\). The effectiveness of the rankings is evaluated using the well-known Normalizing Discount Cumulative Gain (NDCG) metric. The \(\text{NDCG}_n\) for \(n\) retrieved Web services is given by \(^6\) :

\[
\text{NDCG}_n = \frac{\text{DCG}_n}{\text{IDCG}_n}
\]

(4)

Where \(\text{DCG}_n\) is the Discounted Cumulative Gain and \(\text{IDCG}_n\) is the Ideal Discounted Cumulative Gain. The \(\text{IDCG}_n\) is found by calculating the \(\text{DCG}_n\) of the first \(n\) returned Web services. The \(\text{DCG}_n\) is given by :

\[
\text{DCG}_n = \sum_{i=1}^{n} \frac{2^{\text{relevance}(i)} - 1}{\log_2(i + 1)}
\]

(5)

Where \(n\) is the number of Web services retrieved and \(\text{relevance}(s)\) is the graded relevance of the Web service in the \(i\)th position in the ranked list.

The experiments were conducted on a Sony Vaio laptop, with a Core I9 Intel processor, 2.54 GHz clock speed, 8 GB memory RAM. The laptop was running under Windows 7 operating system, and SUN Java Virtual Machine version 1.6. The experimentation has been performed with the number of peers that varies in the range [100-1500], with increments of 200. By generated requests we mean the total number of overall requests produced and forwarded.
on the unstructured P2P network as a consequence of a request submitted to a peer; this parameter depends on the number of peer in the network and on the peers average number of connections to its neighbors.

In particular, our epidemic discovery algorithm and Gnutella P2P Protocol have been compared on the basis of the generated requests and the scalability as the value of peers parameter grows: (i) to prove the better performance of our epidemic algorithm in comparison with Gnutella; (ii) to confirm that the use of the similarity flooding results in an improved scalability. Experimentation results will be analyzed in the following.

The performances results of generated requests evaluation are shown in Figure 2. As the number of peers reached by the request grows, the generated requests increases. On the other hand, we observe the better results obtained by our scalable P2P approach because it uses similarity flooding that decreases the big overhead of network traffic and communication, which gives better accuracy.

To better demonstrate the effectiveness of our scalable P2P approach, a performance analysis has been performed to measure the variation in the execution time according to the number of peer. The execution time has been measured upon the number of invokes that our scalable P2P approach, reflecting the number of discovering SWs in the unstructured P2P network. This helps demonstrate the scalability of our P2P approach. The execution time has been measured using The Eclipse Test and Performance Tools Platform (TPTP5).

The graph in Figure 3 summarizes some of the performance test results. We notice that the execution time increases linearly with the increasing number of peers. The results reported in this figure represent the average execution times found for each given request. Despite the exponential theoretical complexity, the figure 3 shows that the scalable epidemic algorithm can be used, with acceptable execution time. More clearly shown in Figure 3, Gnutella protocol has a very high execution time because it uses simple flooding algorithm. This algorithm leads to high computational time. Our scalable P2P approach performs better than the Gnutella protocol in terms of time execution because it uses similarity flooding that reduces considerably the execution time. The results are very promising.

Fig. 2. Generated Requests and Scalability vs Peer.

Fig. 3. Execution Time vs Peer.

Fig. 4. Matching Quality vs Thresholds.

Fig. 5. NDGC vs Peer and TTL.

5 TPTP: http://www.eclipse.org/tptp/
Figure 4 describes the corresponding curves to the precision, recall and F-measure statistics obtained by applying our scalable P2P approach on the chosen SWs for different thresholds values. As shown in Figure 4, precision increases according to the threshold because high values of thresholds minimize the number of false positives in the P set, what gives higher precision. However, recall decreases when the threshold increases. This is due to the fact that high thresholds reduce the number of true positives in the P set, what leads to low recalls. Using a reduced value of threshold leads to low precisions, and using a high threshold leads to low recalls. Figure 4 confirms this observation and shows poor performances in terms of F-measure for low and high values of thresholds. Figure 4 shows that best results of our approach are obtained with Threshold=0.6, which correspond to the topmost value of the F-measure curve. However, the performance measure of our approach shows that the results returned are quality and good.

The results of the NDGC measure are shown by the graphic of Figure 5. For instance, the NDCG obtained is equal to 0.9 when TTL = 7. We also observed that the value of NDGC slightly decreases when the threshold increases. As shown by this graphic, the results obtained by our approach are very satisfactory. Following these results, and other experiments and previous evaluations, all results obtained by our scalable P2P approach are encouraging, both in terms of performances and effectiveness of the distributed discovery of SWs.

5. Conclusion and Future Work

In this paper, we have discussed an efficient approach, for improving the performance and effectiveness of distributed and cooperative discovery in the unstructured P2P networks. The approach exploits an epidemic algorithm to fulfill the users requirements in order to ensure high recall, further reduce the number of messages exchanged and reduce the execution time for discovering SWs. In this approach, we have proposed a similarity flooding which reduces the search space while keeping an acceptable matching quality level. In particular, it reduces considerably the execution time and message overhead on large-size P2P networks, improves the recall, and maintaining at the same time a good quality of the distributed discovery of SWs in the unstructured P2P network. The results of the experimental evaluation indicate a significant performance gain in comparison to existing approaches. The experimental results show that our efficient approach is able dynamically to adapt to network changes and preserve high levels of recall. In the future work, we will focus our effort optimizing the distributed discovery and reducing the communicational complexity by using optimization techniques such as heuristics and meta heuristics.

References