Peer-to-Peer Technology Usage in Web Service Discovery and Matchmaking*

Brahmananda Sapkota[1], Laurentiu Vasiliu[1], Ioan Toma[2],
Dumitru Roman[2], and Chris Bussler[1]

[1] Digital Enterprise Research Institute, Galway
[2] Digital Enterprise Research Institute, Innsbruck
{firstname.lastname}@deri.org

Abstract. This paper presents a dynamic and scalable mechanism for discovery of semantically enriched descriptions of Web services. By employing Web Service Modeling Ontology (WSMO) as the underlying framework for describing both user requests and Web services, and combining it with the usage of Peer-to-Peer technology in this context, a scalable, distributed, dynamic and flexible discovery mechanism is obtained. A use case scenario is presented for supporting the viability of such a mechanism.

1 Introduction

A combination of Web services and Peer-to-Peer (P2P) technologies have potential to be an effective means for solving business integration problems (e.g. data consistency, discovery, validation, etc) due to their distributed nature and interoperability features. In the research communities, work is underway looking for such a combination [2], [3], [14]. A P2P network of Web service registries has been proposed in [2] whereas [3] provides an infrastructure dealing with P2P scalability issues at the cost of topology maintenance. The P2P approach taken in [2] is vulnerable to a single point of failure as it uses a single central server to index all Web service registries.

In this paper, we propose a combination of Web service and P2P technologies for supporting communication between different Web service platforms such as WSMX [4] and IRS-III [5] thereby addressing the aforementioned problems during service discovery and matchmaking. Such a combination can support scalability, maximize search recall; enable dynamic and distributed Web service discovery and matchmaking at a minimum maintenance cost. In addition, unlike other approaches (e.g., [2], [14], [15]) it avoids the problem of single point of failure. The distributed Web service discovery is supported by a goal decomposition algorithm. The aforementioned goals are validated through a real-life business process integration use case.

The rest of the paper is structured as follows: Section 2 and Section 3 provide a short overview of Web services and P2P technologies. Section 4 gives an insight to the general solution achieved by combining these technologies. Section 5 takes this

---

* This material is based upon works supported by the EU funding under the DIP project (FP6 – 507483) and the ASG project (FP6 – 004617) and by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

M. Kitsuregawa et al. (Eds.): WISE 2005, LNCS 3806, pp. 418 – 425, 2005.

further focusing on service discovery and Section 6 presents a real use case. Section 7 describes related work and Section 8 concludes the paper suggesting future work.

2 Web Services

Web services [17] have added a new level of functionality to the current Web by taking a first step towards seamless integration of distributed software components using web standards. Nevertheless, current Web service technologies around Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL) and Universal Description, Discovery and Integration (UDDI) operate at a syntactic level and, therefore, they still require human interaction to a large extent [7]. This limits the scalability and greatly curtails the added value envisioned with the advent of Web services.

Recent research aimed at making Web content more machine-processable under the common term Semantic Web [18] are gaining momentum, particularly in the context of Web service usage. Here, semantic markup shall be exploited to automate the tasks of Web service discovery, composition and invocation, thus enabling interoperation between them with minimum human intervention. In this context, WSMO [6] aims to describe relevant aspects related to general services, with the ultimate goal of enabling the (total or partial) automation of the tasks involved in both inter- and intra-enterprise integration. In this paper, we commit to the use of WSMO for describing Web services because of the enriched semantic capabilities it provides.

3 Peer-to-Peer Computing

P2P technology is known to be a simple to administer and powerful to compute technology. Functionally, P2P technology comes in three different forms, namely: *pure-P2P*, where each participant has equal role; *hybrid-P2P*, where some of the nodes act as a central server and provide search facilities; and *server based-P2P*, where a dedicated server indexes control information but the data flow is between participants [8].

A P2P system functions through the interaction between ‘neighbours’ thus avoiding central control and the single point of failure. This interaction is defined through the algorithms that enforce the topology and the roles of the participants. Each participant can store/access other participant’s data, if needed, with or without knowing its origin. If a participant fails, other participants can take over rebuilding the system. P2P technology also allows load distributing among participants.

4 Peer-to-Peer Enabled SWS Platforms

In order to discover a service that satisfies a requester's goal, multiple Web services may have to be consulted, since a single Web service may not be able to fully satisfy a requester's goal. Therefore, Web services need to be able to communicate with each other which can be made possible by the use of P2P technology because of their: distributed nature; scalability, flexibility and manageability characteristics; ability to deal with data, protocol, and machine heterogeneity.

4.1 SWS Platforms in Peer-to-Peer Networks

Several Semantic Web Service (SWS) platforms, e.g. WSMX, IRS-III, etc, emerged along with the SWS technology. The interconnection between these SWS platforms is desired and needed to deal with complex business process integration problems. On a small scale, these SWS platforms can communicate by registering themselves with each other as Semantic Web services. However, scalability becomes an issue when the number of these platforms grows. Through the combination of P2P and SWS technologies, these platforms can be discovered dynamically. Since pure-P2P does not scale, we adapt hybrid P2P topology for connecting these platforms together.

Our adapted hybrid P2P network is similar to the Super-node based P2P network as described in [10]. However, in our P2P network nodes are described semantically and the Super-node is a relative concept. Therefore, a node S that is seen as Super-node from neighbour X can be seen as 'normal' node from neighbour Y. The network evolves dynamically electing such Super-nodes and the requests are processed among these Super-nodes enabling load balancing and failure recovery.

4.2 Clustering SWS Platforms

In order for a system to be practically compelling, it needs to be efficient. In the context of this paper, a system is efficient if maintenance cost is low and the system is quick to execute requests. To achieve this efficiency, clustering of SWS platforms based on similar Web service description is proposed in Figure 1. A cluster consists of at least one Super-node. Each cluster is maintained and managed by one of the Super-nodes, which is chosen dynamically based on its availability, processing power, storage capacity, etc. We call this Super-node the cluster *manager*. It indexes the Web services registered with its cluster and facilitates communication within and between clusters. A cluster *manager* can become a member of another cluster.

The following algorithm, inspired by [13], is used for selecting the cluster *manager*, where availability α, available storage ζ, and processing power ω are the metrics used for evaluation of Super-nodes.

```
initialize candidate set to null
for all S in cluster C
  if $S_\alpha \geq (1/n)\Sigma^n S_\alpha$, candidate $\cup$ {$S_\alpha$}
for all K in candidate
  if $K_\zeta \geq (1/n)\Sigma^n K_\zeta$ candidate $\cup$ {$K_\zeta$}
for all L in candidate
  if $L_\omega \geq (1/n)\Sigma^n L_\omega$ candidate $\cup$ {$L_\omega$}
for all M in candidate
return best M.
```

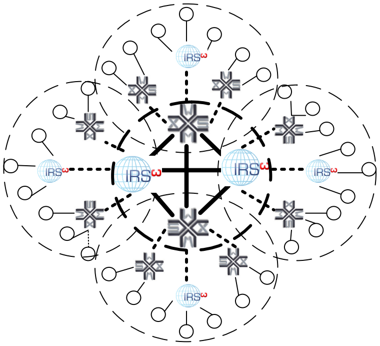

Fig. 1. SWS Platforms in Service Clusters

best implies: highest availability, largest storage, and highest processing power amongst others. If no such node exists, the one with highest availability is chosen.

A cluster may not have a *manager* if all of its members are *manager* of other clusters. Such a cluster is called a *super-cluster*. In Figure 1, this notion is indicated by a thick dotted circle. The user's goals are always executed in the *super-cluster*. This significantly reduces the total number of invocations required to execute a request. When a new member joins the network but no similar service cluster exists, then it creates new cluster, becomes its *manager* and notifies *super-cluster* of its existence.

4.3 Goal Decomposition

The complexity of the user initiated goal can vary from simple to difficult. In the latter case goal decomposition is necessary because of several reasons such as: the information required to evaluate the user goal is distributed; the goal consists of unrelated concepts; it requires mutually exclusive service composition; the goal can be executed only partially; etc. Thus, based on Ontologies O defining the goal G, it is decomposed into sub goals g_i such that $\cup g_i = G$. The following algorithm gives a set of sub goals:

```
initialize sub-goal sg, to null
for each i of c ∈ pc(G)
  for each a of i
    if |elem(a of i)|> minCard(a)
        sg ∪ (createGoal (a of i))
    else sg ∪ (createGoal(i of c))
return sg
```

where,

i = instance,
C = set of Concepts defined in O; $c \in C$,
a = attribute,
sg = set of sub-goals

The operator *pc*(G) returns all the concepts used in G, the operators *elem* (a) and *minCard* (*a*) return all the value elements of attribute *a*, and their required minimum cardinalities respectively, The operation *createGoal* creates sub-goal using the same formalism used in goal G.

4.4 Service Invocation in Clustered Network

In our clustered network of SWS platforms, a backend-application requests for a service through one of the known SWS platforms which then executes it. The invocation is successful if a matching Web service is found locally. It is pending if a no matching Web service is found locally, but some Super-nodes are yet to be consulted. If the invocation is pending, the *manager* of the relevant cluster is consulted in an attempt to successfully resolve the invocation. The result of this successful invocation ends up at the initiating backend-application. If all Super-nodes are invoked and the invocation is still pending, it is deemed unsuccessful. Invocation through Super-nodes thus reduces the number of calls required to satisfy a user's goal.

5 Discovery and Matchmaking in P2P

This section describes how efficient and effective the discovery and matchmaking can be in a hybrid P2P network of SWS platforms. We concentrate on the services provided by a P2P network to the SWS platforms only. Explanation on discovery and matchmaking is out of the scope of this paper but for illustration purposes basic characteristics of both Web service discovery and matchmaking processes are presented.

5.1 Discovery

The process of obtaining a set of services which can possibly fulfill a user request is called Service Discovery. At a conceptual level, we employ the framework presented in [9] as the high level methodology for Web service discovery, which in turn, will make use of a P2P architecture, to gain greater scalability. The Web service discovery is done based on matching abstracted goal descriptions with semantic annotations of web services. The whole P2P architecture will deal specifically with the Web service discovery i.e. *semantic-based approaches* to web service discovery.

In a real-life scenario, there can be a huge number of requests for web services. Processing large numbers of requests in a single SWS platform will introduce the scalability issues. Therefore, the processing task needs to be distributed which can be made possible by connecting SWS platforms in a P2P network. Similarly, knowledge of the availability of a particular Web service will become visible (indirectly) to all participants in the P2P network. It is important to note that service composition may require fulfilling the requester's goal which can be achieved by decomposing the goal, executing these sub-goals at neighbouring SWS platforms, obtaining partial services, and finally composing them.

5.2 Matchmaking

The process of identifying the Web service from a collection of ‘similar’ Web services that ‘fully’ satisfies the requester's goal is called matchmaking. In SWS platforms matchmaking is done by evaluating the user's goal against the capabilities of the Web services registered with these SWS platforms. The mechanism outlined in [9] will be used for the purpose of matchmaking. The process seems simple but is quite complicated in reality. Web services whose capabilities are specified in a language other than English, for instance, might have a better deal but may never be considered. However, such heterogeneity can be resolved by using data mediators.

In the event of matchmaking in the P2P network of SWS platforms, the request will always be evaluated locally first. In the case of partial-match or no-match, the request will be forwarded to the cluster *manager* that will then execute the request in order to find out the local match. While doing so, if the matchmaker component is over loaded, the request will be forwarded to another member of the same cluster. Thus, use of P2P technology in Web service technology will enable load balancing.

6 Case Study

The Super-node based P2P network in our case study is shown in Figure 2. The *manager*s of AirService, TrainService, BusService, and HotelService clusters are forming a *super-cluster*, which is indicated by the centre circle. The clustered network shown in Figure 2 is invisible to the service requester and it is created over time. Therefore, a service requester can contact any of the members of the network to initiate a TravelPlan request.

The goal in this example is that the requester is looking for a cheap round trip holiday plan with the following constraints: the trip originates in Galway and its ends in Bucharest; it should include a visit to and at least one night’s stay in Dublin, Amsterdam and Geneva; the trip should cost ≤ € 1500 including accommodation; the duration of the trip should be ≤ 3 weeks. Following is the WSMO description of this goal:

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-rule"
namespace { "http://example.org/goals#",
 dc   _"http://purl.org/dc/elements/1.1#",
 wsml _"http://www.wsmo.org/wsml/wsml-syntax#",
        loc  _"http://www.wsmo.org/ontologies/location#",
        tr   _"http://example.org/tripPlanOntology"}
goal_"http://example.org/tripPlanGalwayBucharest"
importsOntology { _"http://example.org/tripPlanOntology",
               _"http://www.wsmo.org/ontologies/locationOntology"}
capability
postcondition
 definedBy //User wants a holiday trip
 ?planTrip [ planRequester hasValue ?planRequester,
          item hasValue ?plan ] memberOf tr#tripplan
 and
 ?plan[ trip hasValue ?trip ] memberOf tr#trip
 //A round trip: Galway to Galway with stop over in: Dublin, Amsterdam, Geneva, Bucharest
  and
 ?trip [ origin hasValue loc#Galway,
         destination hasValue loc#Galway,
         stopOvers hasValue {loc#Dublin,loc#Amsterdam,loc#Geneva,loc#Bucharest},
         accomodation hasValue ?accomodation,
         travel hasValue ?travel,
         duration hasValue ?tripDuration ] memberOf tr#trip
 and
 ?accomodation [ numberOfNights hasValue ?numberOfNights,
              location hasValue {loc#Dublin,loc#Amsterdam,loc#Geneva,loc#Bucharest},
              price hasValue ?accomodationPrice ] memberOf tr#accomodation
 and
 ?travel [ transportation hasValue ?transportation,
        price hasValue ?travelPrice ] memberOf tr#travel
 and // The total numebr of nights is thus ≥ 3
 ?numberOfNights ≥ 3
 and //The duration of the trip should be less than 3 weeks/21 days
 ?tripDuration < 21
 and //accommodation + transportation should be less than 1500 euro:
 (?accomodationPrice + ?travelPrice) < 1500
```

Let us follow the execution of the requester's goal submitted to one of the SWS platforms in figure 2. Initially, a goal is presented to the SWS platform number 1. As the goal consists of unrelated sub-goals and the goal receiver is a member of TrainService cluster, the goal cannot be fully discovered and matched. Therefore, the goal is forwarded to its cluster *manager*; numbered 2 in Figure 2. The goal is decomposed by the *manager* of the TrainService using the algorithm defined in section 4.3, and the following sub-goals are obtained: *tripGalwayDublin*, *tripDublinAmsterdam*, *tripAmsterdamGeneva*, *tripGenavaBucharest*, *hotelDublin*, *hotelAmsterdam*, *hotelGeneva*, and *hotelBucharest* where the trip is a round trip. The trip sub-goals are forwarded to the AirService and BusService cluster *manager* and are also evaluated locally. The accommodation-related goals are forwarded to the HotelService cluster *manager*. Every cluster *manager* performs their respective task and returns the result to the *manager* of the TrainService cluster. The results received from other *managers*

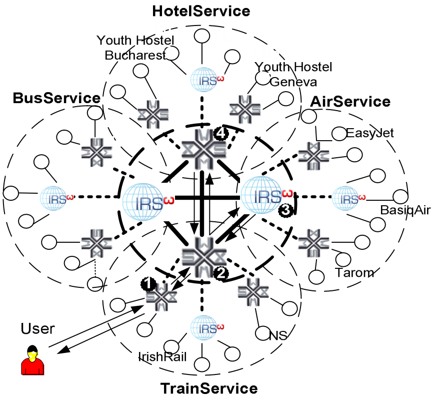

Fig. 2. Service Discovery and Matchmaking

are merged and a set of potential TravelPlan are obtained by TrainService cluster manager. The result is further filtered according to the user constraints to obtain a minimal result set. The cheapest cost plan is selected and is presented to the requester.

7 Related Works

The combination of P2P and Semantic Web service technologies for allowing better service discovery and matchmaking has recently attracted the attention of different research communities. In [14] an interesting solution for combining semantic descriptions of services and P2P technologies is described. This approach is based on DAML-S, for service description and *pure-P2P* as underling network architecture. A DAML-S matchmaker is used to match a request against services registered with one peer. The choice of technology made this approach vulnerable to: a) scalability due to query flooding; b) the use of DAML-S for service description inherits the drawbacks of this model [19]. Our approach is different to [14], as we do not use *pure-P2P* technology and our approach is guided by the much richer WSMO framework.

Semantic technologies are used as well in METEOR-S approach ([2], [15]) where a P2P network of semantically enriched UDDI registries is envisioned. Requests are created by populating a predefined user request template. P2P network of registries envisioned in METEOR-S suffers from single point of failure problem as there is a single entry point for requests to enter the network. In this context, our approach is features co-op between nodes, is faster, and is more scalable. The discovery mechanism presented in [16] also suffers from the same problems because of thee choice of unstructured P2P technology. It limits the discovery scope by using TTL (Time To Live) so the best services may not be discovered. In our Supernode based P2P approach, the discovery of the best service available in the network is possible. A generic algorithm for goal decomposition and for selecting a cluster manager is presented which can be extended or adapted to the requirements of the applications.

8 Conclusions and Future Works

In this paper we presented a mechanism for merging two different technologies that are designed to achieve the same goals. This combination opened up a new horizon in the area of Semantic Web services thereby supporting distributed, dynamic and scalable Web service discovery and matchmaking. We illustrated the use of a load balancing mechanism and scalable co-operation between the SWS platforms.

The example presented in this paper reflected the benefits of this combination by enabling greater load balancing, quicker discovery of Web services, requiring less invocations, and better scalability than other similar approaches. In the next step we intend to further consolidate our work with SWS platforms by providing ‘fully’ semantic description of the functional properties of each node in the network and refine the implementation towards this direction.

References

1. Garofalakis, J., Panagis, Y., Sakkopoulos, E., Tsakalidis, A.: Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?", International Workshop on Web Engineering, Santa Cruz (2004).
2. Verma, K. , Sivashanmugam, K. , Seth, A. , Patil, A.: METEOR-S: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services, Journal of Information technology and Management, (2004).
3. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services, Second International Conference on Peer-to-Peer Computing (P2P '02), (2002).
4. Web Service Execution Environment (WSMX), *http://www.wsmx.org*, (2005)
5. Domingue, J., Cabral, L., Hakimpour, F., Sell, D., Motta, E.: A platform and Infrastructure for Creating WSMO-based Semantic Web Services, In Proceedings of the WSMO Implementation Workshop (WIW 2004), Frankfurt, (2004).
6. Roman, D., Lausen, H., Keller, U.: Web Service Modeling Ontology-Standard (WSMO), WSMO-Working Draft, DERI, (2005). available at: http://www.wsmo.org/TR/d2/v1.2/
7. Alesso, H.P., Smith, C.F.: Developing Semantic Web Services, A.K. Peters, Massachusetts, (2004).
8. Schollmeier, R.: A Definition of Peer-to-Peer Networking for the Classification of Peer-to Peer Architecture and Applications, In Proceedings of the First International Conference on Peer-to-Peer Computing (P2P '01), (2001).
9. Keller, U., Lara, R., Polleres, A., Toma, I., Kifer, M., Fensel, D.: WSMO Web Service Discovery, WSMO Working Draft v0.1, DERI, (2004).
10. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network, In Proceedings of the 19th International Conference of Data Engineering (ICDE), Banglore, (2003)
11. Cimpian, E., Mocan, A., Roman, D., Scharffe, F., Scicluna, J.: WSMO Mediators, WSMO Working Draft v0.1, DERI, (2005).
12. Mocan, A., Cimpian, E.: WSMX Data Mediation, WSMX Working Draft v0.2, DERI, (2005).
13. Sapkota, B.: Design of a Peer-to-Peer Protocol for AmbientDB, Masters Thesis, University of Twente, the Netherlands, (2003).
14. Paolucci, M., Sycara, K. P., Nishimura, T., Srinivasan, N.: Using DAML-S for P2P dicovery. In ICWS, (2003).
15. Patil, A. A. , Oundhakar, S. A., Sheth, A. P. , Verma, K.: METEROR-S Web Service Annotation Framework. In Proceedings of Thirteenth International World Wide Web Conference Proceedings, (2004).
16. Banaei-Kashani, F., Chen, C.-C., Shahabi, C.: WSPDS: Web Services Peer-to-Peer Discovery Service. In Proceedings of the International Conference on Internet Computing, (2004).
17. Alonso, G., Casati, F., Kuno, H., Machiraju., V.: Web Services, Springer, (2003).
18. Berners-Lee, T., Hendler, J., Lassila., O.: The Semantic Web. Scientific American, (2001).
19. Lara, R., Roman, D., Polleres, A., Fensel, D.: A Conceptual Comparison of WSMO and OWL-S. In proceedings of European Conference on Web Services (ECOWS 2004), Erfurt, Germany, (2004).