Matchmaking for Semantic Web Services

Lee Choon Chiat Linpeng Huang Jinkui Xie
Dept. of Computer Science & Engineering, Shanghai Jiao Tong University
Shanghai 200030, PR China
choonchiat@myrealbox.com huang-lp@cs.sjtu.edu.cn jkxie@cs.sjtu.edu.cn

Abstract

The most important topic about the web recently is semantic web services. The current trend of the software development is to reduce the development time, but at the same time increase the functionality, reusability and stability of the building block without worrying about how the service is implemented. As the number of services available on the web increases, the difficulty for both the service requester and provider to look for the most suitable potential counterpart will also increase. Matchmaking therefore is an important aspect of the web interactions, which enable both service requester and provider trim down the amount of time to meet the most suitable potential counterpart. In this paper we describe how to blend the matchmaking algorithm to the typical web service architecture.

1. Introduction

Nowadays, as the rapid grow of the number and variety of services on the semantic web, model that allow service providers and requesters to interoperate, seek and cooperate with each other will be required. Ontology of services called DAML-S, which is part of the DARPA Agent Markup Language program, has been developed to provide a set of basic concepts and relations for declaring and describing services, by utilizing the ontology structuring mechanisms provided by DAML [2]. In addition, with the possibility of interacting with a far greater number of potential counterparts, it enlarges possibilities comes the problem of having to select the best among the tones of available counterparts. And matchmaking is the process that can be used to prune the space of possible matches among compatible offers and requests [4].

A DAML-S service is characterized by three types of knowledge: a service-profile, service-model, and service-grounding. A service-profile provides a high-level description of a service and its provider, it is used to request or advertise services with discovery services and capability registries. Service-model is used to describe in terms of a process model, which details both the control structure and data flow structure of the service required to execute a service of the web service. The service-grounding specifies the details of how to access the service, details mainly to do with protocol and message formats, serialization, transport, and addressing. Generally speaking, the service-profile provides the information needed for an agent to discover a service, whereas the service-model provides enough information for an agent to make use of a service, and the service-grounding describes how agents can communicate with and invoke the service [3].

2. Matchmaking

First of all, before we deep into the requirement of matchmaking, we must explain what are web services? It is quite self-explanatory from the term itself, but it is more than just refer to accessing services over the web. We would like to borrow the definition of web services taken from an IBM web services tutorial for a more proper explanation:

Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. In other words, web services are interoperable building blocks for constructing applications, where the architecture enables applications to connect to other applications. Instead, the current web application enables user as the only party to connect to it.

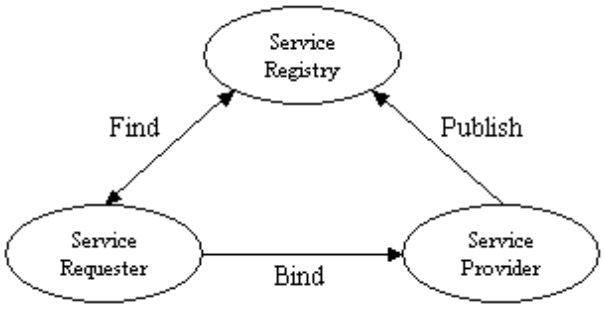

Figure 1

Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04)
0-7695-2225-4/04 $ 20.00 IEEE

Figure 1 is the typical web services architecture. The service registry is the place for service provider to publish their service description. And then, the service requester searches all the interested services available via the service registry, the services description contains all the information for the service requester that enable them to bind to the service provider to use the service.

The process by which parties that are interested in having exchange of useful information and then put in contact with potential counterparts for further cooperation is called matchmaking. Matching together the features that are provided by one party and required by another defines the process of matchmaking accurately. The minimal requirements for the functionalities that a matchmaking service provides are the features of advertising a service, and browsing or querying a repository of advertised services [1].

1. Advertising: Both provider and requester can publish their description of the features of the service for the product that is providing or requesting. Such description is published as an advertisement in the matchmaking service. Some additional information may be added, such as the contact details to the advertisement to make it possible for a potential counterpart to follow up. In addition, Corollary information might be expressed on negotiable terms and condition as well as the rules of engagement for the negotiation process [1].
2. Querying: To search a relevant advertisement among the currently available ones. The submitter can submit a query to filter out all the existing advertisements that are not under the submitter's interest. The query expresses constraints over aspects of advertised services that the submitter only interested in [1].
3. Browsing: Interest party can browse through the currently available advertisement by the browsing feature offered by the matchmaker. The matchmaker also responsible to maintains an advertisement database, where all the posted advertisements are stored. To facilitate browsing, the matchmaking services may provide a classification of adverts and of the terms used in them [1].

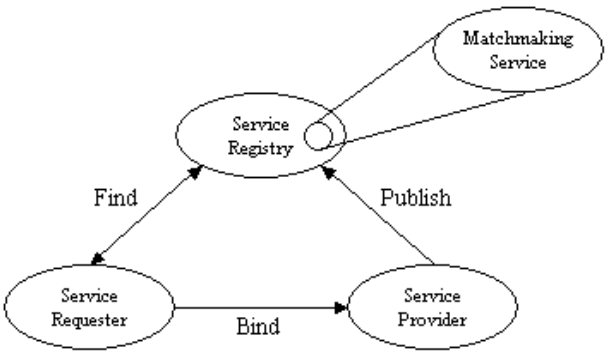

Figure 2

After we have stated our matchmaking service clearly, we have to consider where to put into our matchmaking service in figure 1. Since all the requests are searched via the service registry, it is obvious that the matchmaking service should become the part of the service registry as shown in figure 2.

When the service requester is making a request on the service registry, if they are more than a service provided by a particular service provider, the matchmaking service is then triggered; all the relevant services provided by a particular provided will be displayed.

3. Concept of Matchmaker

The problem of finding and choosing the potential counterparts for service is arising since the rapid grow of service provider available on the web. Apparently, the main problem is not the insufficient number of service provider, or the service provided by the service provider does not meet the requesters' needs, but is because of both the provider and requester unable to find the most suitable potential counterparts. Through the mediation of the matchmaker, which matches service offers with service requests, potential counterparts will be able to find each other, thus fully utilizes the semantic web. Based on some real-life experience, we would rather be able to compare descriptions with different levels of specificity and complexity than use an approach based on exact matching. For example a more general description for the services found, without any restrictions, should match the service required. More specific descriptions should also be matched. Finally, descriptions that are neither more specific nor more general but that describe services that would be compatible with the service required should also match.

4. Matchmaker functionalities

It is easy to imaging that using technique of Description Logics to classify the service descriptions, since Description Logics influences DAML+OIL. Description Logics are a family of knowledge representation formalisms. They are based on the notion of concepts and roles, and are mainly characterized by constructors that allow complex concepts and roles to be built from atomic ones [4]. The basic functionalities of the matchmaking service are as follow:

DL Expressiveness	DL Syntax	Serv. Descript. Lang.
ALC, also called S when transitively closed primitive roles are included	A	Concept
	$\top$	Thing
	$\bot$	Nothing
	$(C \subseteq D)$	Subsumption
	$(C \equiv D)$	Equivalence
	R	Properties
	R	Object Properties
	$(C \bigcap D)$	Conjunction
	$(C \bigcup D)$	Disjunction
	$\neg C$	Negation
	$\forall R.C$	Universal Role Rest.
	$\exists R.C$	Existential Role Rest.
N	$\leq nR.\top$	Non-Qualified Card.
	$\geq nR.\top$	
	$=nR.\top$	
Q	$\leq nR.C$	Qualified Cardinality
	$\geq nR.C$	
	$=nR.C$	
I	R^{-}	Inverse Roles
H	$(R \subseteq S)$	Subsumption of Roles
	$(R \equiv S)$	Equivalence of Roles
O	{o}	Nominals
	$\exists T.\{o\}$	Value Restrictions
(D)	D	Datatype System
	T	Datatype Property
	$\exists T.d$	Exist. Datat. Rest.
	$\forall T.d$	Univ. Datat. Rest.

Table 1

5. Matchmaking Algorithm

The matches for service description S are:
1. equivalent concepts to S;
2. sub-concepts of S;
3. super-concepts of S that are subsumed by the serviceDescription concept;
4. sub-concepts of any direct super-concept of S whose intersection with S is satisfiable.

Let us consider an example of an online retail shop that sell CD and book, in addition, the shop also provides the book rental service. The retail shop owner decides to use the web service on the internet for her web application, which can provide the service for her potential customers to perform the task such as browsing and ordering CD and book online, and also renting book from her web site.

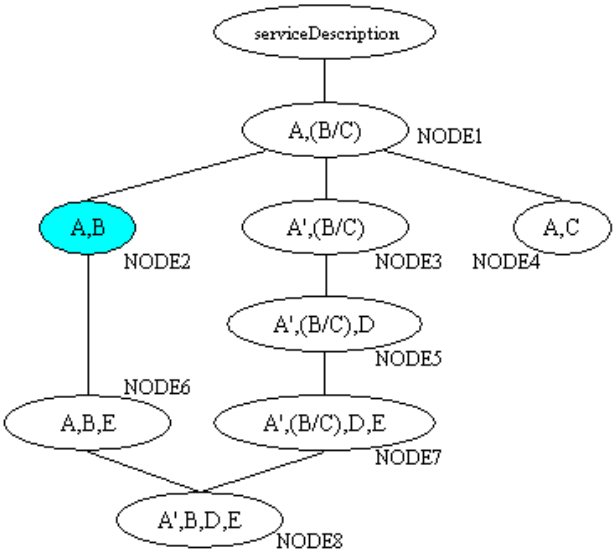

Figure 3

To understand the algorithm, again we use the online retail shop as an example. Assume A can perform the book ordering task, and B can perform the book list browsing task. On the other hand, X can perform the task for CD ordering, and Y can perform the task for CD list browsing. Finally, i is the service which provide the book rental service. The possibility to find a service that match the exact requirement would be very low, but if finding each task individually would also be time consuming, in addition, with the possibility that each individual service could not be found. The better idea would be to combine related tasks, such as A and B, X and Y, and i, and find it as a whole. Considering a service provider provides some services as shown in figure 3, which can be found by any interested party. Consider finding the service provider which provides tasks A and B. The query is denoted as NODE2 in figure 3. We evaluate sequentially the four propositions of the algorithm. In our example, there is no equivalent concept to NODE2. NODE6 and NODE8 are sub-concepts of NODE2 and as such are marked as matches. The third step is to look for super-concepts of NODE2 up to the serviceDescription node. Hence, NODE1 is marked as a match. Finally, the last step of the algorithm gives us the nodes NODE3, NODE5 and NODE7. If A' is the compliment to A (i.e. more advance version of A, which is not only can perform the book ordering task, but also include the book rental service.) which can perform task A and i, then the service provided at NODE3 will be more suitable to the requirement, and this will also save the time to look for service which perform task i. Finally, after we found the service which provides tasks X and Y, we will then be able to combine the two services from different providers. The composition is beyond the scope of our current research.

6. Matchmaking Architecture

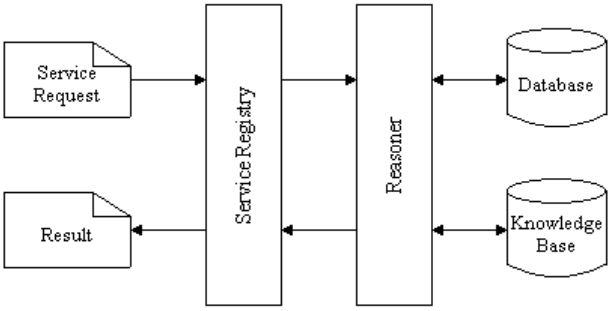

Figure 4

Figure 4 depicts the architecture of the matchmaking for semantic web service. First the requester submit a request to the service registry, the reasoner which is residing in the service registry will then retrieve all the advertisement from database and all the relevant rules from the knowledge. All the matched advertisements will be displayed as a result back to the requester.

7. Future Work

FaCT system is developed by Ian Horrocks; it is a Description Logic classifier that can be used for modal logic satisfiability testing. This system contains two reasoners, one for the logic SHIQ, and another one for the logic SHF. The reasoner provides XML syntax for the definition of ontologies and the CORBA interface, which makes the reasoner available as a service for other application to use. Some of the features which is needed is not included in the reasoner, thus we have to develop a reasoner system that meet our requirement.

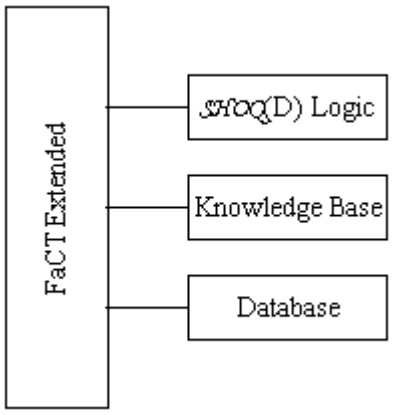

Figure 5

Figure 5 shows the reasoner that we are to develop. The reasoner is based on the FaCT system, but we use the SHOQ(D) logic instead of the SHIQ and SHF logic in the current FaCT system. A database is needed to store the large amounts of information of the advertisement. Since advertisements can be added, removed and modified, the knowledge base system is necessary to be included in the reasoner; it is used to reclassify the advertisement after any modification has been done in the database.

8. Conclusion

Service requests are compared with service advertisements through a subsumption based inferencing mechanism using DAML ontologies. A variety of different approaches have been used to match agent advertisements and service requests. The algorithm shown is just one of the way might be used for matchmaking. Base on the research we have done so far, we are very optimistic and believe that automated semantic web service will be achieved sooner or later in the future.

9. Acknowledgement

This research is supported by Shanghai grid e-institute project 200308.

10. References

[1] David Trastour, Claudio Bartolini and Javier Gonzalez-Castillo, A Semantic Web Approach to Service Description for Matchmaking of Services, HP Labs Technical Report.

[2] Terry R. Payne, Massimo Paolucci, Katia Sycara, Advertising and Matching DAML-S Service Descriptions, Semantic Web Working Symposium (SWWS), 2001.

[3] Anupriya Ankolekar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David Martin, Drew McDermott, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry Payne, Katia Sycara, DAML-S: Web Service Description for the Semantic Web, International Semantic Web Conference 2002: 348-363.

[4] Javier Gonzalez-Castillo, David Trastour, Claudio Bartolini, Description Logics for Matchmaking of Services, HP Labs Technical Report.

[5] Katia Sycara, Matthias Klusch, Seth Widoff, Jianguo Lu, Dynamic Service Matchmaking Among Agents in Open Information Environment, ACM SIGMOD Record, Special Issue on Semantic Interoperability in Global Information Systems, A. Ouksel, A. Sheth (Eds.), 28(1):47-53, 1999.