A OWL-Based Semantic Web Service Discovery Framework

Xinqi Wang   Xueli Yu

Dept. of computer science and tech

Taiyuan Univ. of Tech.,

Taiyuan, Shanxi, P. R. China

d12s34f57@hotmail.com

Abstract. Current Semantic Web Service discovery mechanism consumes too much resource and lacks the ability to be used in real-world, large-scale situations. This paper propose a new kind of Semantic Web Service framework based on the concept of dividing the Semantic Web into different domains, and by introducing a kind of Semantic Web Service Router to help reduce the set of service description information which service request information has to be matched against., we hope the service discovery process can proceed more quickly and more precisely and the speed of service discovery can be reduced .

1. Introduction

Semantic Web is proposed by Tim Berners-Lee [1], it is a promising vision that is based on the idea that adding machine-understandable semantic information to web resources will facilitate automation of many tasks, with the advent of semantic web. Great emphasis has been put on the discovery and composition of Semantic Web Services (SWS). Under the current SWS discovery mechanism, if a service publisher wants its service to be discovered and invoked by the service requester, the service publisher would have to register the description information of its service to a centralized registry (such as UDDI). The shortcoming of this approach is that the service requester would have to send its service requesting information to the service registry, and let the service registry match the service requesting information against the description information of the service stored in the service registry in order to find out whether there is a published service that can satisfy the service request. This approach has been adopted by many applications, but most of the applications just provide service to a specific organization (such as a company or students in college campus), in another words, current SWS is like the earliest computer network, such as the ARPANET, in order to make Semantic Web and SWS mainstream, whose service can be accessed by almost anybody, just like the current internet, great scalability has to be achieved.

To resolve the problem of current SWS's limited scalability (services are mostly provided to limited amount of people and within limited geographic area), this paper presents a new kind of SWS framework, which draws some lessons from current Internet architecture [2] and TCP/IP [3] protocol. This new framework uses ontology to realize the concept of Semantic Web Service Router and by registering different domain of SWS to different SWSR, the size of the set of service description

information which service requesting information has to match against can be reduced, the service discovery process can proceed more quickly and more precisely.

2. Description of the Framework

In the SWS framework proposed in this paper, service description information is not stored in centralized registry, but in Semantic Web Service Router (SWSR) distributed across the network, unlike the traditional router, the SWSR is not employed to realize the communication among heterogeneous networks, but to reduce the set of service description of the published SWS the service request has to be matched against.

The upper layer of SWSR represents more general domain and the lower layer of SWSR represents more specific domain. Every one of SWSRs of the upper layer represents several sub-domains, in the lowest layer of SWSR, different SWSRs stores the description information of the SWS from different domains, the upper layer of SWSRs store the information, such as IP address, of the different SWSRs of the lower level, the main job of the higher level of the SWSRs is to help service requester find out the lower layer SWSR which stores the service description information of the SWS requested by the service requester.


```
         ┌────────┐
         │SWSR_A* │  ◄------------  Level n
         └───┬────┘
             ┊
         ┌───┴────┐
         │ SWSR_A │  ◄------------  Level 2
         └───┬────┘
        ┌────┴─────┐
   ┌────┴───┐  ┌───┴────┐
   │ SWSR_b │  │ SWSR_a │  ◄------------  Level 1
   └────────┘  └────────┘
```
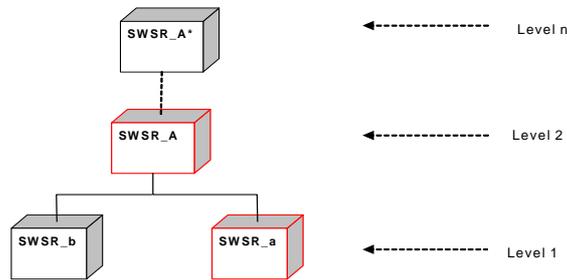
Fig.1

As shown in Fig.1, In level 1 of SWSRs, lets assumes SWSR_a stores the service description information of SWS from the domain a, SWSR_b stores the service description information of SWS from the domain b, and there is SWSR_A in level 2,which is one level higher than the SWSR_a and SWSR_b, the SWSR_A represent the domain A, domain A subsume domain a and domain b, which is the domain of SWSR_a and SWSR_b respectively. And SWSR_A stores the description information of SWSR_a and SWSR_b (such as IP address). Through SWSR_A, the service requester interested in SWS of domain a will be able to find SWSR_a , because SWSR_a stores all the SWS description information about SWS from domain a, from SWSR_a, service requester can find the actual server providing the SWS it's interesting in. There are also SWSRs which represent domains more general than domain A represented by the SWSR_A, these SWSRs reside on higher level of

SWSRs, such is the SWSR_A* in level n, they can help service requester find the location of the SWSR that more specifically subsumes the domain of SWS requested by the service requester.

Under the framework, the service publisher registers the SWS description information to the SWSR representing the domain, which the SWS belong to. The packet sent by the service requester would first go to the SWSR in service requester's local network, if the local SWSR does not represent the domain which matches or subsumes the domain of the requested service, then the packet would be redirected to the upper level of SWSR which represent the super-domain of the domain of the current SWSR, the process repeat itself until a SWSR matching or subsuming the domain of the requested service is found or the packet reaches the highest level of SWSR, such as SWSR A* in Fig.1,this SWSR, through use of semantic matching algorithm, would try to find ,among the sub-domains subsumed by the domain of the current SWSR, the sub-domain which matches or subsumes the domain of the SWS requested by the service requester. If there is no such sub-domain, which means there are no SWS currently available that can directly satisfy the service request. Now service-matching algorithm can be used to examine whether several services can be composed together to satisfy the service request, if the services cannot be found, then return failure. We would discuss the application of service composition in our framework in the future work. This paper would focus on the overall mechanism of the proposed framework.

In the discussion above, if the sub-domain which matches or subsumes the domain of the SWS requested by the service requester is found among the domains subsumed by the domain of the SWSR, and because the SWSR knows the address of the lower level layer of SWSRs which domain is subsumed by the domain of higher level SWSR, the higher level SWSR would redirect the data packet sent by the service requester to the corresponding lower level SWSRs found in the domain matching process, the lower level SWSR would do similar domain matching process after receiving the packet, if a sub-domain were found which would subsume the domain of the requested service more specifically, the packet would again be redirected to the corresponding SWSR, in which the some domain matching algorism would be executed, the process runs repeatedly until finally the packet is sent to the SWSR in which the domain of the SWSR and the domain of the requested service are proved to be exact match and in this SWSR, the actual SWS description information is stored, in this SWSR, the description information of requested SWS would be matched against the description information of the published SWS stored in this SWSR, if match were found, the SWSR would redirect the packet to the server in which the SWS is provided, therefore, the contact between the service requester and the service publisher is established ,the service discovery process is completed.Fig.2 shows the service discovery process:
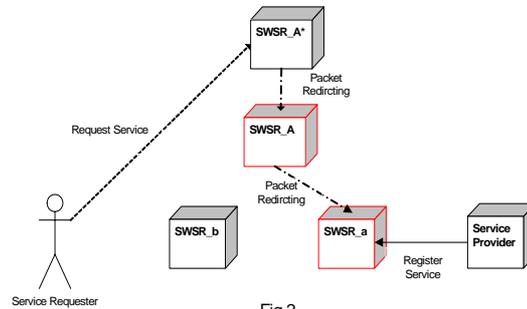
Fig.2

In the framework mentioned above, the end users only need to provide the Semantic Web with the description information of their requested service, the service description includes the description of the domain which the requested service belongs to, and the description of the actual requested service, such as input, output, pre and post-conditions. If there exists a published service which can satisfy the request, the network would discovery this service automatically, and establish the contact between the end user and the service provider, all the underlying network architecture detail, such as the IP address would be transparent to the users.

As described in above paragraph, our proposed new framework divides Semantic Web into different layers and domains, so that the set of published services which service request has to match against can be reduced, as we make it clear here: we are not saying that the current SWS framework would fail in doing the task of service discovery and service matching, but because in the current SWS framework, service description information from different domains of SWS may be stored in the same repertory, so the service request may has to match all the descriptions information stored in the repertory, even the SWS does not remotely relate to service request, therefore, times and resources have been wasted when service request has to match against service description information of a totally unrelated SWS, in other words, the set of published which service request has to match against may includes many published SWS from domains different from the domain of service request, so the time and resources consumed during the service matching process under current framework is bigger than that under our proposed framework. We hope that, as Semantic Web expands, our proposed framework can save time and resources in the service discovery and matching process, and make expanded Semantic Web more usable, thus improve the scalability of Semantic Web and SWS.

3. Reason to choose OWL

In the second part of our paper, we propose a new kind of SWS framework; the central idea of our SWS framework is to divide the Semantic Web into different layers of domains and register different SWS according its domain, so service requester only needs to have its request matched against the domain of SWS which subsumes its requested service most specifically, therefore the size of the set of service description of published SWS service request has to be matched against can

be reduced. To achieve that goal in practice, it is very important to let all the participant of the process of the service discovery process to share the same knowledge of the dividing and layering of the Semantic Web, and because it has some nice properties for knowledge sharing among AI software (e.g., semantics independent of reader and context), we decided to use ontology to model the dividing and layering of the Semantic Web. The OWL web ontology modeling language is a W3C standard, comparing with other ontology modeling languages, OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes, and because those quality of OWL, it can better describe the relationships between different domain of Semantic Web, so we choose OWL as the ontology modeling language to model the layering and dividing of the Semantic Web. The detail description of OWL can be found at [4], We will discuss in detail the realization of our SWS framework using OWL in our future works.

4. Conclusion and Future Works

In this paper, we have presented a new kind of SWS framework, through this new framework; the concept of dividing the Semantic Web into different domain of SWS has been realized. And we propose the new SWS framework in order to reduce the size of the set of service description of published SWS the service request has to be matched against, and consequentially, we hope that by introducing the framework, the SWS discovery can proceed more quickly and more efficiently.

In this paper, we have not discussed the SWS composition in our framework and the context awareness is not discussed in detail. Both of those two issues will play major roles in the future service-oriented Semantic Web we envisioned, therefore, in our future research, we will have to incorporate the SWS composition into our proposed SWS framework and build a shared context ontology to help realize context awareness in our proposed framework, by doing that, we hope that our SWS framework can be further improved and become more practical in real-world applications.

References

1. W3C: Soap version 1.2 part 0: Primer, June 2003. http://www.w3.org/TR/2003/REC-soap12-part0-20030624/.
2. Yale University: Introduction to TCP/IP, http://pclt.cis.yale.edu/pclt/COMM/TCPIP.HTM.
3. PROTOCOLS.COM: TCP/IP Reference Page, http://www.protocols.com/pbook/tcpip1.htm.
4. World Wide Web Consortium: OWL Web Ontology Language Overview, Feburary.2004, http://www.w3.org/TR/owl-features/.