A CAN-BASED P2P INFRASTRUCTURE FOR SEMANTIC WEB SERVICES

Nizamuddin Channa , Shanping Li , Wei Shi and Gang Peng
College of Computer Science, Zhejiang University Hangzhou, P.R.China 310027

Abstract: After merger of Web Services and Semantic Web, Semantic Web Services (SWS) has received a lot of attention from researchers due to its ability of automatic Web Service discovery, execution and composition. Currently Web Service systems, which publish WSDL-described Web Services in UDDIs, cannot support SWS and UDDI has become the bottleneck of the whole system and would cause single node failure problems. Therefore, we propose a CAN-based P2P system to replace traditional UDDI, by distributing the functions of the UDDI among all the peers in the P2P network. At the same time, we design an ontology-based mechanism, guaranteeing every service would be registered on a specific peer in the CAN-based P2P network, according to the service's ontology. By replacing the UDDI, our system improves the scalability and stability of the SWS system, and realizes an efficient ontology-based discovery of Semantic Web Services.

Key words: Semantic Web Service, P2P network, CAN, Ontology-based discovery

1. INTRODUCTION

The term Semantic Web Services (SWS) [1] has received much attention from researchers due to its ability of automatic Web Service discovery, execution and composition. The Semantic Web encompasses efforts to build

The research, is funded by the Natural Science Foundation of China (No. 60174053, No. 60473052)

a new WWW architecture that enhances content with formal semantics. That means, content is made suitable for machine consumption, as opposed to content that is only intended for human consumption. This will enable automated agents to reason about Web content, and produce an intelligent response to unforeseen situations. By describing Web Services in contents to AI inspired markup languages, such as DARPA Agent Markup Language (DAML) [7] and Web Ontology Language (OWL) [8], Semantic Web Services supports automatic service discovery, execution, composition and interoperation, while in traditional Web Service systems, all the Web Services described in Web Services Description Language (WSDL) [9] are published in the Universal Description Discovery & Integration (UDDI) [2], which acts as a central server. As a consequence of these fragile central servers, Web Service systems are vulnerable in front of malicious attacks, and cannot easily scale to support a large number of Web Services. Moreover, traditional UDDI-Based Web Service systems lack support of Semantic Web Services, which are described in AI-inspired markup languages. Therefore, it addresses a need to propose a new Web Service system, which may be more scalable and stable as compared with traditional Web Service systems and provides SWS support gracefully.

In recent years P2P computing [10] has emerged as a novel and popular model of computation and gained significant attention from both industry field as well as academic field. P2P network models are becoming popular for information sharing and data exchange. These models offer important advantages of decentralization and scalability by distributing capacity and load among all the peers in the network. And they have been treated as an alternative to traditional Clients/Servers infrastructure in many areas.

In this paper, we propose three-layered novel system architecture to support SWS and enhance the scalability and stability of Web Service systems, which uses a Content Addressable Network-Based (CAN-Based) Peer-to-Peer (P2P) network as its infrastructure, to replace traditional Web Service systems. And the whole system is composed of P2P Infrastructure, Web Service Distributor (WSD) and Web Service Translator (WST). To publish Web Service Distributor Language for described Web Services, the Web Services would be translated into OWL-Services (OWL-S) [4] by WST firstly (In case the Web Services have been described in OWL, obviously, there is no need to submit Web Services to the WST for translation). After the translation, according to the ontology contained in OWL-S, WSD would register and publish these Web Services on specific peers in the P2P infrastructure at last. By organizing the Web Services according to our proposed architecture, Web Service discovery can be realized efficiently. Moreover, our system supports “Vague” Web Service lookup. In contrast to traditional Web Service systems, our proposed system improves the

scalability and stability of the SWS by distributing the Semantic Web Services among all the peers in the P2P infrastructure. Moreover, the Web Service Translator entitles our system to publish existing numerous WSDL-described Web Services.

The rest of this paper is organized as follows: Section 2 presents the overview of our system and Section 3 describes the design and implementation of our system in detail. Section 4 surveys the related work. Section 5 draws the conclusions and describes our future work.

2. SYSTEM OVERVIEW

The framework of our proposed system is shown in Figure 1, which is made up of three major components including Web Service Translator (WST), Web Service Distributor (WSD) and CAN-Based P2P Network. Web Service Translator interprets WSDL-described Web Services into OWL-S. And Web Service Distributor is in charge of distributing OWL-S among peers in the P2P network according to the ontology OWL-S contains. The P2P network functions as the infrastructure of whole system, which replaces the UDDI in traditional Web Service systems. When we publish a Web Service, WST would translate the WSDL files of this Web Service into OWL-S files, and the WSD extracts the ontology contained in OWL-S files and allocates the Web Service to a specific peer in P2P infrastructure according to this ontology. Due to the delicate design of our system, it is easy to realize Web Service discovery efficiently. Because after obtaining the ontology of requested Web Service, we can figure out the coordinates of the Web Service and route to the specific peer quickly according to the CAN [3] protocol.

For automatic registering and discovering services, we need semantic description but currently on the web, a large number of Web Services are available, which are described in syntactic description. Therefore in our system, we use translator, which translates WSDL files into OWL-S and provides semantically enriched description. It is important to note here that until now, this translation process is not functioning fully automatically.

After the translation from WSDL files to OWL-S, the OWL-S must be published for future operations. Traditionally, Web Services would be published on UDDI. On the other hand our proposed system uses P2P network to replace UDDI, in which every peer works as a small UDDI server and cooperates with other peers. The distributor is in-charge of mapping an OWL-described Web Service to a specific peer according to the ontology in the OWL files.

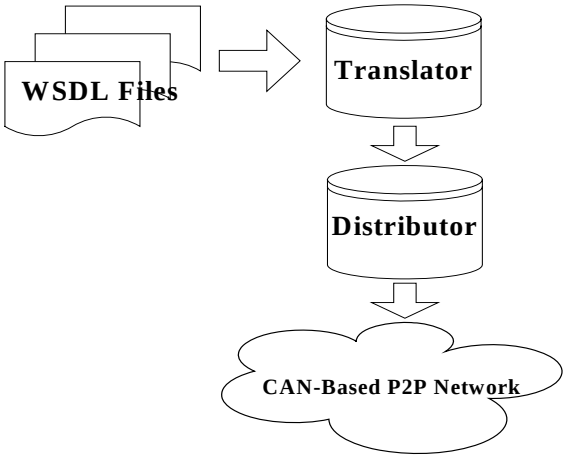

Figure 1. Framework of our system

To replace UDDI with P2P network, we have to resolve the problems of choosing an appropriate protocol to organize the P2P infrastructure. It is also important that we must guarantee that a requested Web Service, which has been published, would be definitely discovered in a lookup process; on the other hand, this lookup process should be finished efficiently and quickly. Based on these requests, we choose CAN-Based P2P network as our infrastructure, because in CAN-Based P2P network, all the resources are well organized according to their coordinates and will be quickly located if their coordinates are available

3. DESIGN AND IMPLEMENTATION

3.1 Translator from WSDL to OWL-S

The translator lies on the first layer of our system, which translates WSDL files into OWL-S files. OWL-S are ontology services which support simple as well as complex services. They enable us to automatically discover, invocate, compose, interoperate and monitor Web Services. OWL-S descriptions are organized into three conceptual areas. These are Service Profiles, Process Models and Service Groundings. The Service Profiles tell what Web Service do? Service Profiles provide general description of a Web Service for advertising and discovering. They provide a way to

describe Web Services offered by the providers and those needed by the requesters. Process Models present how the Web Services work? The Process Models have a number of inputs, outputs, preconditions and effects. Here inputs and outputs specify the data transformation produced by the process. Inputs represent the information, which is required for the execution of the process; outputs are the information that the process returns after its execution. Preconditions specify conditions that must be satisfied for the Web Services to be executed correctly. The effects describe the actual results as consequence of such execution. The Service Grounding specifies the details of how to access Web Services.

The Figure 2 shows the whole structure of translator, which provides an automatic or semi-automatic generation of OWL-S specifications starting from WSDL specifications. This translation generates OWL ontologies, which need to be mapped to existing ontologies in the Semantic Web to be useful for automatic process composition. The outputs of this translator provide the basic structure of an OWL-S description of Web Services and save a great deal of manpower.

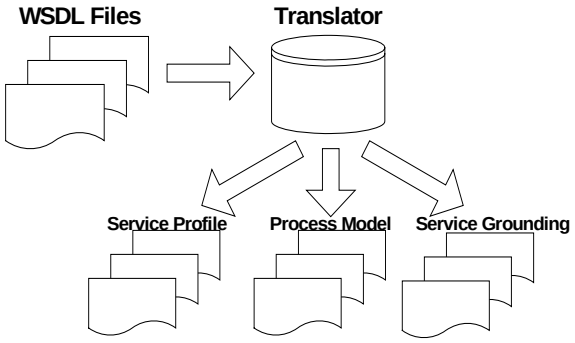

Figure 2. Function of web Service Translator

The translation is based on the following two principles.

1. A WSDL operation is equivalent to an atomic process of the OWL-S Process Model. This observation provides the basic mapping between WSDL and OWL-S. It is used for both the generation of the basic Process Model and for the generation of the Grounding.
2. OWL-S descriptions make use of OWL concepts to specify the content of inputs and outputs, while WSDL makes use of XML Schema data (XSD) types to specify inputs and outputs. Since the OWL-S Grounding

that specifies the mapping between OWL-S Process Models and WSDL does not provide any mapping from concepts to types, therefore we are forced to assume a 1:1 correspondence between them. The second rule generates the basic data used by OWL-S.

Translator's generating OWL-S description from WSDL description can be semi-automatic or manual due to information difference between OWL-S and WSDL-described Web Services. WSDL files provide only input and output information, while OWL-S files contain more than input and output information. Therefore the rest of the OWL-S profile, such as preconditions and effects, must be set manually

3.2 Web service distributor

The Web Service Distributor lies in the second layer of our system, which distributes and publishes Web Services on the P2P infrastructure based on the ontology the Web Services contain.

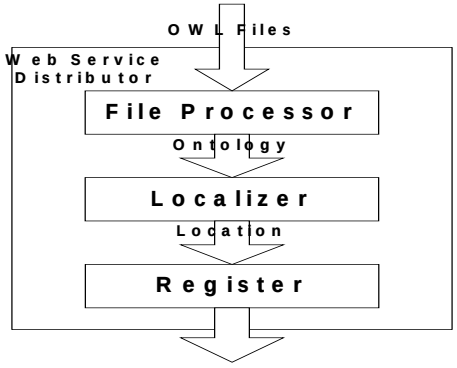

Figure 3. Structure of Web Service Distributor

Its input is in form of three OWL files, which are the outputs of the Web Service Translator. After extracting ontology from these OWL files, it figures out the specific location of the published Web Service in the P2P network. At last, according to this location information, Web Service Distributor would register the published Web Services on a specific peer.

As Figure 3 shows, Web Service Distributor is composed of three layers, including File Processor, Localizer and Register. File Processor takes charge of extracting ontology from OWL files and sends them to Localizer. Possibly, there may be a lot of ontology in OWL files. File Processor only

takes out the ontology in the areas we concern. For example, in our proposed system, we use Input, Output, Precondition and Effects (hereafter IOPEs) to identify Web Services. So, the concerned areas are IOPE, which means File Processor only need to take out the ontology in IOPE. Based on the ontology input, Localizer works out the locations of Web Services according to some location algorithm. This paper, proposes a location algorithm, which establishes a coordinate space by numbering all the possible ontology and figure out the coordinates by mapping the input ontology with this coordinate space. We use an example to simplify our presentation. Figure 4 shows the service ontology and domain ontology. We provide reserve, check in and check out services, and the Room domain is composed of Single Room, Standard Room and Suite. Our location algorithm numbers all of these ontology and form a coordinate space, whose coordinates are 6-bits long. So, a "reserve Single Room" operation would be mapped to the coordinate of "100100" and a "check out suite" operation would be mapped to the coordinate of "001001". For Web Services, different location algorithms would generate different locations. Because of our module design of Distributor, when applying a different location algorithm, we only need to modify the Localizer, not the whole Web Service Distributor.

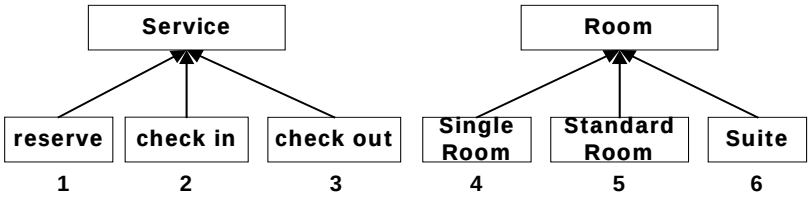

Figure 4. Service ontology and Domain ontology

The bottom of Web Service Distributor is Register, which registers Web Services on specific peers with the help of the locations provided by Localizer. The Register is closely connected with the CAN-Based P2P of our system. Here, we resort to an example to present the whole function of Distributor. Suppose we use IOPE to identify Web Services, which mean we only care about the ontology in input, output, precondition and effect areas, and use the service ontology and domain ontology in Figure 4 when realizing mapping from ontology to coordinates. We also suppose that service's input area contains the operation of "reserve Single Room", service's precondition area contains the operation of "check out Standard Room" and service's effect area contains the operation of "check in Single Room". Based on the areas we concern (IOPE), we build a three-dimension

space, which is shown in Figure 5. According to the location algorithm mentioned above, we can get the coordinates of the service, (100100, 010010, 001100). After obtaining these coordinates, we can easily find out which peer this service should be registered on in the CAN-Based P2P Infrastructure according to CAN algorithm.

In fact, our system does not set up any constraint on the number and order of dimensions. Usually, the areas that users care about are far more than IOPE. Users can establish their own multi-dimension spaces at their wills.

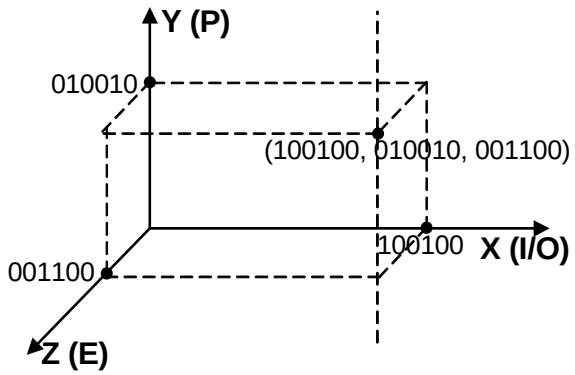

Figure 6. Web Service Register Process

After arranging all the Web Services according to our proposed architecture, Web Service discovery and composition can be realized easily. Referring to the example mentioned above, we use IOPE areas to identify Web Services, which means when requesting a Web Service we must provide the information about IOPE areas. With the help of Web Service Distributor, we can figure out the coordinates of the requested Web Service easily. Based on these coordinates, we can find out the right peer, which contains the requested Web Service. In addition, our system supports “vague” Web Service lookup. For example, the requester can only provide information about Input, Output and Effect areas. After the processing of Distributor, we achieve the X dimension and Z dimension coordinates of requested Web Service, which forms a vertical “line” in Figure 5. Although we still cannot figure out the peer containing requested Web Service. It is obvious that the requested Web Service must be located on one of the peers, whose spaces cover this vertical “line”. As a result, we do not broadcast the Web Service request, but only send the request to these peers, which reduce lookup area greatly.

3.3 CAN-Based P2P Infrastructure

We use CAN protocol to arrange our P2P infrastructure. Every peer in the infrastructure takes charge of a range of space of the whole multi-dimension space. Every Web Service would be registered on the peer, which covers the location of this Web Service.

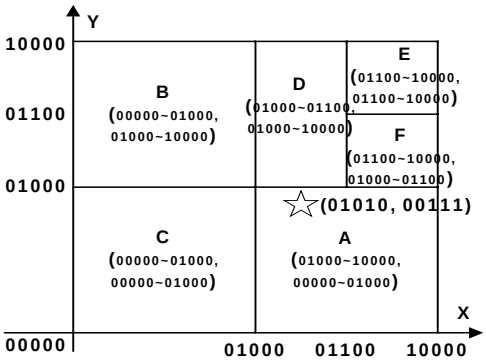

Figure 6. Web Service Register Process

Figure 6 shows an example of how the Web Services are registered on peers of CAN-Based P2P infrastructure. A, B, C, D, E and F are peers, which are in charge a specific area of space separately (rectangles shown in Figure 6). A published Web Service (the star shown in Figure 6), which locates in (01010, 00111), should be registered on the peer A, for its coordinate lies in the area, which peer A is in-charge of.

4. RELATED WORK

Current Web Service systems are based on “central” UDDI servers and syntactic description of Web Services, which are fragile and unscalable, and do not support Semantic Web Services. It is necessary to replace this UDDI structure with a new one. P2P is such an infrastructure, which allows the sharing of resources and services by direct interaction between equal nodes. Until now, there are many P2P systems, such as Napster [11], FreeNet [12] and Gnutella [13], and also appear some excellent P2P resource lookup algorithms, such as Chord [14] and CAN.

In fact, there have appeared some projects, which are using P2P network to replace UDDI and support Semantic Web Service, such as METEOR-S WSDI [5].

METEOR-S WSDI "uses an ontology-based approach to organize registries, enabling semantic classification of all Web Services based on domains". The P2P infrastructure in METEOR-S WSDI project is not a pure P2P network, in which there is a Gateway Peer controlling access to the P2P network and some Auxiliary Peers providing Registries Ontology. These special peers are the performance bottleneck of the P2P infrastructure and make the whole system fragile facing malicious attacks.

As for service ontologies, there also exist some projects, which use service ontologies to describe the semantics of Web Services. Ruoyan Zhang et al. [15] describe an Interface-Matching Automatic Composition technique to generate complex Web Services automatically by capturing user's expected outcomes with the help of service ontologies. But this project is still based on UDDI server, not a more stable infrastructure, such as P2P network. Moreover, it does not propose a method to handle existing numerous traditional Web Services. Mario et al. [6] proposed an ontology-based P2P infrastructure for SWS, which is based on a hypercube P2P topology network. This system is similar with our system, but it does not provide a WSDL to OWL-S translator to handle existing numerous WSDL-described Web Services. Moreover, this system only supports one-dimension coordinates in the P2P network, while our system supports multi-dimension space.

5. CONCLUSIONS & FUTURE WORK

In this paper, we proposed a novel Semantic Web Service system, in which we resorted to CAN-Based infrastructure to replace UDDI and distribute Semantic Web Services among all the peers in P2P infrastructure based on the ontology contained in Semantic Web Services. Because there is no "central server", our system avoids "single node failure" problem, which improves the stability of the whole system. And the P2P infrastructure also makes our system more scalable than traditional Web Service systems by distributing the system function among all the peers, not focusing on only one or a few servers. By abstracting ontology contained in SWS and allocating SWS according to the ontology, our system also provides a promising support for Semantic Web Service.

Now, we have established our SWS system and realized the publishing and discovery of SWS, while the whole life cycle of SWS includes not only Web Service publishing and discovery, but also automatic Web Service

composition and execution. In the future, we would work on realizing auto-composition and auto-execution in our system.

REFERENCES

[1] McIlraith, S., Son, T.C. and Zeng, H. ``Semantic Web Services" , IEEE Intelligent Systems. Special Issue on the Semantic Web. 16(2):46--53, March/April, 2001. Copyright IEEE, 2001.

[2] Universal Description, Discovery and Integration: UDDI Technical White paper. 2000. http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

[3] Sylvia Ratnasamy et al. A Scalable Content-Addressable Network. In Proceedings of the ACM SIGCOMM, 2001.

[4] Owl-S: Semantic Markup for web services. OWL-white paper http://www.daml.org/services/owl-s/1.0/owl-s.pdf

[5] Kunal Verma, etc. “METEOR–S WSDI& A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services”, Journal of Information Technology and Management, Special Issue on Universal Global Integration, Vol. 6, No. 1 (2005) pp. 17-39. Kluwer Academic Publishers

[6] Mario Schlosser, etc. “A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services”, Second International Conference on Peer-to-Peer Computing (P2P'02), September 05 - 07, 2002, Linköping, Sweden

[7] The DAML homepage: http://www.daml.org/

[8] M. Dean et al., “Web Ontology Language (OWL) Reference Version 1.0,” World Wide Web Consortium (W3C) note, Nov. 2002;www.w3.org/TR/2002/WD-owl-ref-20021112.

[9] E. Christensen et al., “Web Services Discription Language (WSDL) 1.1,”World Wide Web Consortium (W3C) note, 2001; www.w3.org/TR/2001/NOTE-wsdl-20010315.

[10] R. Schollmeier, “A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications.” Proceedings of the First International Conference on Peer-to-Peer Computing, 2001.

[11] The Napster homepage, http://www.napster.com

[12] Ian Clarke et al. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Lecture Notes in Computer Science. 2000

[13] http://www.infomaticsonline.co.uk/News/1134977 Informatics Online Web Site,

[14] Ion Stoicay et al. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications 2001.

[15] Ruoyan Zhang, I. Budak Arpinar, Boanerges Aleman-Meza, Automatic Composition of Semantic Web Services , The 2003 International Conference on Web Services (ICWS'03), Las Vegas, Nevada, June 23-26, 2003.