SPiDeR: P2P-Based Web Service Discovery*

Ozgur D. Sahin, Cagdas E. Gerede, Divyakant Agrawal, Amr El Abbadi, Oscar Ibarra, and Jianwen Su

Department of Computer Science,
University of California at Santa Barbara, Santa Barbara, CA 93106
{odsahin, gerede, agrawal, amr, ibarra, su}@cs.ucsb.edu

Abstract. In this paper, we describe SPiDeR, a peer-to-peer (P2P) based framework that supports a variety of Web service discovery operations. SPiDeR organizes the service providers into a structured P2P overlay and allows them to advertise and lookup services in a completely decentralized and dynamic manner. It supports three different kinds of search operations: For advertising and locating services, service providers can use keywords extracted from service descriptions (keyword-based search), categories from a global ontology (ontology-based search), and/or paths from the service automaton (behavior-based search). The users can also rate the quality of the services they use. The ratings are accumulated within the system so that users can query for the quality ratings of the discovered services. Finally, we present the performance of SPiDeR in terms of routing using a simulator.

1 Introduction

The adoption and evolution of the Web services technology continue to happen in many different domains from business environments to scientific applications. This technology promises to enable dynamic integration and interaction of heterogeneous software artifacts, and thereby, to facilitate fast and efficient cooperation among the entities in cooperative environments. Lately, there has been a lot of attention drawn to this promising technology from both industry and academia and it has been supported with various emerging standards and proposals such as SOAP[1], WSDL[2], BPEL[3], and OWL-S[4]; accompanying technologies such as IBM’s Web Sphere, Microsoft’s .NET, and Sun’s J2EE; and several research efforts (see recent conferences such as [5, 6, 7, 8]).

Web services are “software applications identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols”[9]. The main research challenges services oriented computing poses include automated composition, discovery, invocation, monitoring, validation and verification[10]. Service discovery, in particular, refers to the problem of how to search for and locate

* This research was supported in parts by NSF grants CNF 04-23336 and IIS 02-23022.
services, the descriptions of which are usually considered lying in well-defined service repositories.

Recently, a substantial progress has been done in this area thanks to several research and industrial efforts including UDDI registries [11,12], similarity search [13], the query languages and indexing efforts [14,15,16], peer-to-peer (P2P) discovery techniques [17,18,19,20,21], semantic web approaches and ontological matching [22,23]. These solutions, however, are typically limited for 2 reasons:

1. They are usually centralized where there is a single central server (e.g., UDDI registry) that keeps track of all available services. Centralized approach has well-known limitations. It is not scalable since the server has to keep information about all services and answer all queries. It is not fault tolerant because the server is a single point of failure and if the server goes down, the whole service discovery mechanism becomes unusable.
2. They usually offer limited search capabilities. There are different techniques to increase the accuracy of service discovery including functional matching (what a service does), behavioral matching (how a service performs), semantic matching (the underlying semantics of a service) and ontological matching (how a service relates to other services). Each of these provides a different metric to measure the relevance among different services and therefore, each one is important. Many existing approaches, on the other hand, concentrate on a single one or a small subset of these techniques.

In this paper, we address above issues by introducing SPiDeR, a P2P based Web service discovery framework that supports a rich set of search operations. A subset of the participating service providers (those that have good resources) are dynamically assigned as super peers and organized into a structured P2P system. Due to its P2P based design, SPiDeR distributes the tasks of indexing available services and resolving service lookups among the participants, thus providing decentralization, scalability, dynamicity, and fault tolerance. It supports 3 different types of search operations based on keywords, global service ontology, and service behavior. It also has a reputation system component for assessing the quality of the services based on the experiences of other services. The ratings given to the services are stored in the system so that users can lookup for service quality ratings when deciding which of the discovered services to use.

The rest of the paper is organized as follows. The related work is surveyed in Section 2. Section 3 introduces SPiDeR, a P2P based distributed Web service discovery framework. Section 4 describes how the different types of discovery operations (keyword-based search, ontology-based search, and behavior-based search) are supported in the framework. The quality rating scheme that enables the ranking of discovered Web services is also explained in that section. In Section 5, dynamic peer operations are discussed in detail. Those include installing and refreshing service advertisements, performing composite lookups, and indexing at the super peers. Additionally, an evaluation of SPiDeR in terms of routing performance using a simulator is presented in Section 5.4. Finally, the last section concludes the paper and outlines the future work.