An Architecture for Restful Web Service Discovery Using Semantic Interfaces

José Renato Villela Dantas, Universidade de Fortaleza, Fortaleza, Brazil
Pedro Porfirio Muniz Farias, Universidade de Fortaleza, Fortaleza, Brazil

ABSTRACT

In the internet environment, web services based on representational state transfer (REST) have become the de facto standard. The addition of semantics is intended to enhance the description of web services with information that enables automatic agents to understand their data. However, the existence of different languages to semantically describe services makes it difficult to discover and select the service that best meets a requirement. Furthermore, relatively few proposals have a RESTful service semantic description, making the discovery process for RESTful services more difficult. This work proposes a RESTful semantic web service discovery architecture based on semantic interfaces (SERIN). SERIN is an ontology with annotations that semantically describe RESTful web services. This architecture enables software agents to automatically discover and make service calls in order to execute a determined task.

KEYWORDS

Restful Web Service, Semantic Interface, Semantic Web, SERIN, Web Service Discovery

1. INTRODUCTION

REST-based (or RESTful) web services (Fielding, 2000) are widely adopted on resource-oriented architectures (ROA). They bring advantages such as scalability and general interfaces, and they separate the server implementation from the client’s perception of resources (Richardson & Ruby, 2008). Furthermore, they do not obligate the provider to develop heavy service descriptions, as WSDL does for SOAP services.

Considering the increasing amount of available web services, the discovery of an appropriate one that agrees with the user’s requirements is a challenge (Pakari, Kheirkhah, & Jalali, 2014). Web service discovery is the process of identifying the service that best matches some user-provided characteristics.

Most of the web service discovery research has focused on three features (Priyadharshini & Gunasri, 2013; Dong et al., 2013; Patil and Gopal, 2011): (i) matching user requirements with the service description, (ii) the existence of a centralized repository where all service providers must register their services, and (iii) the adoption of semantic web technologies to enrich the web service descriptions.

The service matchmaking (feature (i)) is considered the most complex and difficult task in the discovery process. Moreover, given de dynamic nature of available data, this problem should be solved automatically (Cerón-Figueroa, 2017). The matching task usually considers some degree of matching (Paolucci, Kawamura, Payne, & Sycara, 2002) as an indicator of the similarity between

DOI: 10.4018/IJSWIS.2020010101

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
requirements and service descriptors. As discussed by Dong, Hussain, and Chang in their semantic web service matchmakers survey (Dong, Hussain, & Chang, 2013), many researchers have tried to solve the discovery problem by optimizing the matching degree. These methods essentially compute the matching degree by reducing the distance between the user goals and the service descriptions in the minimum computation time. Usually, the service is described as a function of its inputs, outputs, preconditions, effects, and nonfunctional properties. The discovery process usually tries to match the service descriptor with the requirements descriptor. Some approaches propose to use logical reasoning (Fang, 2018) or clustering (Bukhari, 2018) to organize similar web services and, thus, to improve the performance of web service discovery.

The main problem with the matching is that it is always inaccurate. The consumer establishes the minimal degree of matching that is acceptable to the task. However, this degree of matching is not precise and may lead to errors. Finding a service that has the same (or similar to some degree) inputs, outputs, preconditions and effects is not a sufficient condition for selecting a service. Furthermore, considering the service composition context, the concatenation of services multiplies the imprecision of the matching degree, leading to a very high imprecision at the end of the service chain. For example, consider a chain with five services, each with a matching degree of 95%. The final accumulated matching degree for the whole chain will be $0.95^5 = 0.77$

The imprecision of the matching also increases when multiple ontologies are used to annotate the user requirements or multiple ontologies are used to describe an existent service. (Fellah, 2016)

The existence of a centralized repository (feature (ii)) clearly contributes to a faster discovery process. In most research, the starting point for discovering a service is a registry server, e.g., a UDDI (OASIS, 2005) server. Nowadays, some web service portals, such as the aforementioned ProgrammableWeb and several open government data portals, like https://www.data.gov.uk and https://data.gov.uk, are being adopted as tools to search for web services and APIs. All of them suffer from the same difficulty: the service provider must register its service to make it known. This obligation limits the discovery process to a single point of search, restricting the discovery process to the registered services in the repository. Such a limitation removes all unregistered web services from the discovery process.

Many studies (Dong et al., 2013; Pakari et al., 2014; Nacer & Aissani 2014) have found that the addition of semantics to web service descriptions (feature (iii)) provides a valuable contribution to the discovery task. Unfortunately, the heterogeneous nature of several service description languages complicates the discovery process, because the consumer must know and create a proper service client that can handle each language specification.

Adopting semantic web technologies, Muniz et al. (Muniz, Chaves, Campos Neto, Dantas, & Farias, 2011) proposed the semantic interfaces (SERIN) specification to describe semantic web services in functional terms. Conceptually, SERIN is an abstract interface that is analogous to those in the object-oriented paradigm. The abstract interface defines the elements that the service provides, but it does not concretely describe the service. Based on this abstract interface, service providers implement the web services as concrete representations. The distinction between an abstract interface and a concrete web service interface is important in enabling a more efficient and simpler service discovery. It also enables an exact service matching.

In practice, SERIN is an annotated ontology that provides a mechanism to markup web service functionalities and for locating service endpoints. As an ontology (Gruber, 1993) is, by definition, shared knowledge, such an interface provides a web service description that is known to both the client agent and the service provider.

Using the SERIN specification, it is possible to build an architecture to discover web services from providers that implement one or several semantic interfaces. The discovery architecture identifies the hosts that implement the services. This architecture considers clients to know the services already, because they are described by the semantic interfaces. The scenarios and Motivation for SERIN Discovery was presented in (Dantas, Lira, De Azevedo Muniz, Nunes, & Farias, 2015) using an early
MERA: A Musical Entities Reconciliation Architecture Based on Semantic Technologies
[www.igi-global.com/article/mera/189764?camid=4v1a](www.igi-global.com/article/mera/189764?camid=4v1a)

Abstractive Summarization: A Hybrid Approach for the Compression of Semantic Graphs
[www.igi-global.com/article/abstractive-summarization/153668?camid=4v1a](www.igi-global.com/article/abstractive-summarization/153668?camid=4v1a)