Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 321 (2016) 89–112

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Case-based Reasoning for Web Service Discovery and Selection

Alan De Renzis, Martin Garriga, Andres Flores, Alejandra Cechich[1,2]

GIISCo Research Group
Faculty of Informatics, UNComa University
Nuequen, Argentina

Alejandro Zunino[3]

ISISTAN Research Institute
UNICEN University
Tandil, Buenos Aires, Argentina

Abstract

Web Service discovery and selection deal with the retrieval of the most suitable Web Service, given a required functionality. Addressing an effective solution remains difficult when only functional descriptions of services are available. In this paper, we propose a solution by applying Case-based Reasoning, in which the resemblance between a pair of cases is quantified through a similarity function. We show the feasibility of applying Case-based Reasoning for Web Service discovery and selection, by introducing a novel case representation, learning heuristics and three different similarity functions. We also experimentally validate our proposal with a dataset of 62 real-life Web Services, achieving competitive values in terms of well-known Information Retrieval metrics.

Keywords: Web services, Service Selection, Service Discovery, Case-based Reasoning, Service Oriented Application.

1 Introduction

Service-Oriented Computing (SOC) has seen an ever increasing adoption by providing support for building distributed, inter-organizational applications in heterogeneous environments [14]. Mostly, the software industry has adopted SOC by using Web Service technologies. A Web Service is a program with a well-defined interface that can be located, published, and invoked by using standard Web protocols [5].

[1] This work is supported by projects: PICT 2012-0045 and UNCo–Reuse(04-F001).

[2] Email: {alanderenzis,martin.garriga,andres.flores,alejandra.cechic}@fi.uncoma.edu.ar

[3] Email: alejandro.zunino@isistan.unicen.edu.ar

http://dx.doi.org/10.1016/j.entcs.2016.02.006

However, a broadly use of the SOC paradigm requires efficient approaches to allow service discovery, selection, integration and consumption from within applications [29]. Currently, developers are required to manually search for suitable services to then provide the adequate "glue-code" for their assembly into a service-oriented application [9]. Even with a wieldy candidates list, a skillful developer must determine the most appropriate service for the consumer application. This implies a prohibitive effort into discovering services, analyzing the suitability of retrieved candidates (i.e., service selection) and identifying the set of adjustments for the final integration of a selected candidate service.

In this work we make use of Case-based Reasoning (CBR) [1]– from the Artificial Intelligence (AI) field – to overcome the aforementioned problems in Web Service Discovery and Selection. A Case-based Reasoner solves problems by using or adapting solutions from old recurrent problems [35]. Sometimes called similarity searching systems, the most important characteristic of CBR systems is the effectiveness of the similarity function used to quantify the degree of resemblance between a pair of cases [25].

Our proposal models a Case-based Reasoner for Service Selection, where the main contribution is threefold. We define a case representation capturing information in Web Services functional descriptions (typically WSDL). Moreover, we draw a parallel among the key steps in CBR and the problem of Web Service Discovery and Selection. Finally, we provide three implementations for the similarity function, concerning structural and semantic aspects from functional service descriptions.

The rest of the paper is organized as follows. Section 2 details the service selection process. Section 3 presents the application of CBR in the context of service selection. Section 4 details the alternatives for the similarity function. Section 5 presents the experimental validation of the approach. Section 6 discusses related work. Conclusions and future work are presented afterwards.

2 Service Selection

During development of a Service-oriented Application, some of the comprising software pieces could be fulfilled by the connection to Web Services. In this case, a list of candidate Web Services could be obtained by making use of any service discovery registry. Nevertheless, even with a wieldy candidates' list, a developer must be skillful enough both to determine the most appropriate service and to shape the adaptation artifacts for seamless integration of the selected service. Therefore, a reliable and practical support is required to make those decisions. For this, in previous work [12,16] we defined an approach for service selection.

The service selection method is based in an Interface Compatibility assessment of the candidate Web Services and the (potentially partial) specification of the required functionality – depicted in Figure 1. The procedure matches the required interface I_R and the interface (I_S) provided by a candidate service S (previously discovered). All available information from the two interfaces is gathered to be assessed at semantic and structural levels. The semantic assessment makes use