A Computational Logic Application Framework for Service Discovery and Contracting

Marco Alberti, New University of Lisbon, Portugal
Massimiliano Cattafi, University of Ferrara, Italy
Federico Chesani, University of Bologna, Italy
Marco Gavanelli, University of Ferrara, Italy
Evelina Lamma, University of Ferrara, Italy
Paola Mello, University of Bologna, Italy
Marco Montali, Free University of Bozen-Bolzano, Italy
Paolo Torroni, University of Bologna, Italy

ABSTRACT

In Semantic Web technologies, searching for a service means identifying components that can potentially satisfy user needs in terms of inputs and outputs (discovery) and devise a fruitful interaction with the customer (contracting). In this paper, the authors present an application framework that encompasses both the discovery and the contracting steps in a unified search process. In particular, the authors accommodate service discovery by ontology-based reasoning and contracting by reasoning about behavioural interfaces, published in a formal language. To this purpose, the authors consider a formal approach grounded on Computational Logic. They define, illustrate, and evaluate a framework, called SCIFF Reasoning Engine (SRE), which can establish if a Semantic Web Service and a requester can fruitfully inter-operate, by computing a possible interaction plan based on the behavioural interfaces of both. The same operational machinery used for contracting can be used for runtime verification.

Keywords: Abductive Logic Programming, Analysis, Contracting, Interoperability, Proof Procedure, Rules, Semantic Discovery, Web Services

DOI: 10.4018/jwsr.2011070101
INTRODUCTION

Service Oriented Architecture (SOA) and Web services are emerging as standard architectures for distributed application development. Eventually, the use of off-the-shelf solutions/services is becoming possible, although concerns about the adoption of such components have been raised. In particular, the search of services on the basis of the functionality they provide, rather than on some syntactical property, is still an open research issue. Some authors are seeking for a possible solution to this problem among the technologies for the Semantic Web (McIlraith et al., 2001; Kifer et al., 2004). The idea is to augment Web service descriptions by semantic information that can be used to search for Semantic Web Services (SWS).

WEB SERVICE DISCOVERY AND CONTRACTING WITH SRE

In our view, searching for a service means to identify components that i) can potentially satisfy the user needs, and ii) can be invoked by the customers and interact with them. Of course, an interaction is successful if it satisfies user/service goals and constraints; as an example, a user might not want to provide a credit card number to a non-certified service, or a service could disallow credit card payments for items out of stock or with more than 30% discount. Hence, a user request should contain not only a description (given in semantic terms) of the user desires, but also the user constraints about the content and the order of the exchanged messages, to be matched with the constraints that constitute a “behavioural interface” of the user/service. We consider the search of a SWS as the process of selecting, among a given set of services, those components that both i) satisfy the ontological requirements, by providing the requested functionality; and ii) satisfy the constraints on interaction, by supporting the requested behaviour.

In this article we present SRE (SCIFF Reasoning Engine), a framework for searching Semantic Web Services that takes into account requested functionalities as well as requested behaviours. Following Kifer et al. (2004), SRE adopts a two-step search process (Figure 1). For the first phase, called discovery, it extends a well-known algorithm from the literature (Paolucci et al., 2002). In particular, it considers a requester’s desires, and, using ontology-based reasoning on knowledge expressed in OWL (Bechhofer et al., 2004), produces a shortlist of services that can potentially satisfy a request of such a kind. The second step, called contracting, matches the requester’s behavioural interface with those of each shortlisted service. The purpose is to establish constructively whether an interaction can be effectively achieved, and if such interaction leads to achieve the user/service goals. Our choice has been to represent behavioural interfaces using a declarative, rule-based approach, and to exploit computational logic techniques to perform the reasoning task. Note that “contract” is a term also used in other contexts, such as in software engineering (Design By Contract, Brunel et al., 2004). This work does not focus on software engineering issues, and we use the terms “contract” and “contracting” in the sense it is used by others in the SWS literature.

We formalise the external behavior interfaces of users and web services in a declarative language which is a modification of the SCIFF abductive logic programming language (Alberti et al., 2008), originally developed for the specification of open societies. In this new language, behavioural interfaces are defined by Integrity Constraints (ICs): a sort of reactive rules used to generate and reason about expectations on possible evolutions of a given interaction. The SCIFF language is equipped with a proof procedure, which SRE exploits to automatically reason upon the behavioural interfaces. Such a reasoning task aims to establish if an interaction can be effectively achieved and, in case of a positive answer, to provide also a sort of a (partial) plan of a possible interaction.

In previous work (Alberti et al., 2007), we presented a prototype of the SRE framework,
Request-Driven Cross-Media Content Adaptation Technique
www.igi-global.com/chapter/request-driven-cross-media-content/43649?camid=4v1a