We propose a framework and tractable algorithms for semantic-based automated Web service composition, fully compliant with Semantic Web technologies. The approach exploits the recently proposed Concept Abduction inference service in Description Logics to extend Concept covering definition to expressive logics, and to solve Concept Covering problems in a significant subset of OWL-DL. We show how the proposed approach also deals with not-exact solutions, computing an approximate composition and providing an explanation of which part of the request is not covered by the composite service. We present the formalization of the approach, the proposed algorithms, a prototype system implementing the approach, and illustrate experiments carried out with it.

Keywords: concept abduction; concept covering; description logics; knowledge-representation; OWL-S; semantic-based discovery

INTRODUCTION

Web service composition amounts to the orchestration of a certain number of existing Web services (or Web service providers) to provide a composite service able to satisfy the user’s requirements, in case a single Web service is not adequate. Various approaches have been recently proposed able to provide the orchestration of a Web services set, based on different aspects of both Web service (Fensel & Bussler, 2002; Motta, Domingue, Cabral, & Gaspari, 2003; Sirin, Hendler, & Parsia, 2003) and composition flow (Korhonen, Pajunen, & Puustjarvi, 2003; Laukkanen & Helin, 2003; Mecella, Parisi Presicce, & Pernici, 2002; Narayanan & McIlraith, 2002) modeling. But,
due to the lack of automatic procedures able to cope with nonexact matches, in many of such approaches there is the need for a human intervention in order to establish candidate execution flows.

Full automation of Web services discovery and orchestration requires as a first step providing service descriptions—what a service offers—that have to be well-defined, machine understandable, and processable.

The Semantic Web initiative aims to provide Web resources with a meaning, with the aid of formal languages and ontologies to allow reasoning on service description. Such type of Web services are now named Semantic Web services: “a Semantic Web service (SWS) is a Web service whose description is in a language that has well-defined semantics. Therefore, it is unambiguously computer interpretable and facilitates maximal automation and dynamism in Web service discovery, selection, composition, negotiation” (Sycara, Paolucci, Ankolekar, & Srinivasan, 2003); therefore the Semantic Web provides—or should provide—the infrastructure for the semantic interoperability of Web services (Cabral, Domingue, Motta, Payne, & Hakimpour, 2004).

The approach we propose here models the discovery, retrieval, and composition with respect to a resource retrieval scenario; that is, a scenario where there is a request and several available resources, which can satisfy the request in a one to one match (one resource retrieved for the request) or in a one to many match (several retrieved resources whose combination satisfies the request). Our target model is hence a user-oriented and friendly framework, where a typical request is, for example, “I’d like to book a hotel provided with a swimming pool and a fitness center” rather than only “Effects = HOTEL RESERVATION” and a typical service description is like “We book for you hotels near the sea provided with all the facilities: swimming pool, fitness center, children area, and restaurants” rather than only “Preconditions = VALID CREDIT CARD; Effects = HOTEL RESERVATION.” What we would like to point out is that limiting the information provided by services to only preconditions (and/or inputs) and effects (and/or outputs) makes hard to detail the service provided. We believe that a service should provide two kinds of information: (1) description of the service behavior, even not a functional one; (2) execution information:

1. **Descriptive information:** Description of the service behavior, even not a functional one. A simple, free text description is not enough if you want to describe in a machine understandable way, that is, in the Semantic Web vision, such behavior and its details.

2. **Execution information:** Effects. How the Web service changes the state of the domain world—(Outputs. What it changes in the domain world). Preconditions. Which state of the world is needed to make the Web service producing such change—(Inputs. What it needs to change the state of the world) (Ghallab, Nau, & Traverso, 2004).

Composed Web services can be atomic ones (having a single step of execution) or composite services themselves. The service composition process can hence create services with an increasing level of complexity, built out of services at a lower abstraction level. Obviously, service composition is not based on the physical integration of its components (Alonso, Casti, Kuno, & Machiraju, 2003): Each service has, in fact, an interface through which it can be invoked, and the base components of the service remain so separated from the composite service. The main issues in service composition are hence modeling—how to define the process model—and execution—the middleware that executes the process model.

In this article, we propose a framework and algorithms, fully compliant with the Semantic Web vision and its technologies, for the automated discovery and composition in a Semantic Web-services orchestration scenario.
Related Content

Karma2: Provenance Management for Data-Driven Workflows
www.igi-global.com/chapter/karma2-provenance-management-data-driven/41528?camid=4v1a

A Dependable Infrastructure for Cooperative Web Services Coordination
www.igi-global.com/chapter/dependable-infrastructure-cooperative-web-services/66953?camid=4v1a
Authorization Service for Web Services and its Application in a Health Care Domain
www.igi-global.com/article/authorization-service-web-services-its/3071?camid=4v1a

A Scalable Multi-Tenant Architecture for Business Process Executions
www.igi-global.com/article/scalable-multi-tenant-architecture-business/70388?camid=4v1a