Applying CBR Over an AI Planner for Dynamic Web Service Composition

Fouad Henni, LIO Laboratory Oran, University of Mostaganem, Algeria
Baghdad Atmani, LIO Laboratory Oran, University of Oran, Algeria

ABSTRACT

Web services have emerged as a major technology for deploying automated interactions between distributed and heterogeneous applications. The main advantage of Web services composition is the possibility of creating value-added services by combining existing ones to achieve customized tasks. How to combine these services efficiently into an arrangement that is both functionally sound and architecturally realizable is a very challenging topic that has founded a significant research area within computer science. A great deal of recent Web-related research has concentrated on dynamic Web service composition. Most of proposed models for dynamic composition use semantic descriptions of Web services through the construction of domain ontology. In this paper, we present our approach to dynamically produce composite services. It is based on the use of two Artificial Intelligence (AI) techniques: Case-Based Reasoning (CBR) and AI planning. Our motivating scenario concerns a national system for the monitoring of childhood immunization.

Keywords: Artificial Intelligence (AI) Planning, Case-Based Planning (CBP), Case-Based Reasoning (CBR), Dynamic Composition, Monitoring Childhood Immunization, Semantic Web Services

INTRODUCTION

Defined by the W3C, a Web service is a software component identified by a URL, whose public interfaces and bindings are defined and described using XML. Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. This has led to the emergence of Web services as a standard mechanism for accessing information and software components programmatically (Agarwal et al., 2008, p. 1).

Web service interfaces are described by means of Web Services Description Language (WSDL) files that specify how a service can be called. The communication between different services is based on the standard Web protocols, usually HTTP. Web services exchange XML messages that are encapsulated with the Simple Object Access Protocol (SOAP) and can be offered and discovered by the Universal Description Discovery and Integration (UDDI) that provides a mechanism to clients to find services.

DOI: 10.4018/ijitwe.2013100103
One of the advantages of Web services technology is a high reusability of already existing functionalities that can be easily accessed with open Web service standards. The development of new processes and applications requires less time, effort and competencies. Additionally, the flexibility is increased, because new services can be easily integrated and used services easily replaced.

Service composition refers to the technique of composing arbitrarily complex services from relatively simpler services available over the Internet. Composition of Web services enables businesses to interact with each other and facilitates seamless business-to-business or enterprise application integration. Applications are to be assembled from a set of appropriate Web services and no longer written manually (Srivastava & Koehler, 2003, p. 28).

For example, a composite Web service for an online order from a retailer Web site could bring together a number of internal and external services such as credit checking, inventory status checking, inventory update, shipping, etc.

Web service composition is currently one the most hyped and addressed issue in the Service Oriented Computing. Several models, techniques and languages have been proposed to achieve service composition.

The construction of a composite Web service can be made up in three main steps (not necessarily in this order): (a) Creation of the process model specifying control and data flow among the activities. (b) Discovery, selection and binding of concrete Web services to every activity in the process model. (c) Execution of the composite service by a coordinating entity (e.g. a process execution engine) (Sivasubramanian et al., 2009, pp. 1-3).

In order to classify composition strategies, two service composition characteristics are examined: (a) Type of service discovery/selection/binding: it can be done either at design time (early binding) or at runtime (late binding). (b) Degree of automation in the creation of the process model; it can be manual, semi-automated or fully automated.

In static composition the process model is created manually and the bindings of concrete Web services to the process activities are done at design time. Semi-dynamic composition strategies actively support the user with the creation of the process model and/or in the services selection and bindings. Finally, in Dynamic composition the creation of the process model and the services selection and bindings are made at runtime.

In this paper, we aim to show evidence that dynamic Web service composition can be achieved using jointly two AI techniques: Case-Based Reasoning (CBR) and AI planning. A planner is used to generate dynamically a new composition plan from scratch. However, plan generation algorithms are known to be time consuming. So CBR is used as a way to memorize past experiences in order to reuse or adapt a previous successful composition solution to solve an actual user query. As a result, a solution can be provided quickly. The approach is applied to a national research project for the monitoring of childhood immunization.

The remainder of this paper is organized as follows: the second section presents the main ideas in dynamic composition of Web services and our motivating scenario. The third section deals with the foundations and principles of our approach which consists on the use of CBR and AI planning. The architecture of our system and the implementation of the motivating scenario are presented in the fourth section. The paper is concluded by a discussion of the solution, some limitations, and future works.

BACKGROUND

Dynamic Composition of Services

In dynamic composition, automated tools are used to analyze a user query, and select and assemble Web service interfaces so that their composition will solve the user demand. From a user perspective, the composite Web service will continue to be considered as a simple service, even though it is composed of several Web services.
Mining for Web Personalization
Penelope Markellou, Maria Rigou and Spiros Sirmakessis (2005). *Web Mining: Applications and Techniques* (pp. 27-49).

[www.igi-global.com/chapter/mining-web-personalization/31132?camid=4v1a](www.igi-global.com/chapter/mining-web-personalization/31132?camid=4v1a)