Automatic Dynamic Web Service Composition Using AND/OR Directed Graphs

Hajar Elmaghraoui, Mohammed V University in Rabat, Ecole Mohammadia d’Ingénieurs, Rabat, Morocco
Laila Benhlima, Mohammed V University in Rabat, Ecole Mohammadia d’Ingénieurs, Rabat, Morocco
Dalila Chihadmi, Mohammed V University in Rabat, Ecole Mohammadia d’Ingénieurs, Rabat, Morocco

ABSTRACT

In this article, the authors propose a dynamic web service composition approach based on representing the semantic relationship between web services using a weighted directed AND/OR graph. The nodes in this graph represent available services while the arcs represent the semantic input/output dependencies among them. The novelty of this work consists of constructing the graph and computing offline the shortest paths between each pair of its nodes to disconnect this tedious task from the composition query process. A set of dynamic optimization techniques has been included to reduce the size of the graph and thus improve the scalability and performance of this approach. In addition to the sequence and fork relations between services, this solution also supports the parallel relation. Furthermore, a recovery mechanism is integrated to ensure the continuity of the execution of the composition.

KEYWORDS

AND/OR Graph, Discovery, Failure Recovery, Matchmaking, Web Service Composition

1. INTRODUCTION

Web services are defined by the World Wide Web Consortium (W3C) as a software system designed to support interoperable machine-to-machine interactions over a network. They provide mechanisms that enable the combination of multiple services to create new business processes. Despite the large number of services already published, the current needs of business enterprises are very important and, in most cases, cannot be served by a single web service. However, setting up new services to satisfy the changing business needs is a time-consuming and expensive task. To meet these business requirements, there is a growing need for web services composition to build complex web services by simply reusing existing available services. A web service composition is the combination of a set of available services from different providers to satisfy a customer requirement that cannot be achieved by any existing single service. Such a composition provides a new value-added service called composite
service. The biggest benefit of web service composition is to save time and development efforts by reusing the existing services instead of creating a new one from scratch. The variety of composition techniques can be classified into two main aspects (Dustdar et al., 2008):

- **Static and Dynamic composition**: The first aspect concerns the moment when the service is selected to be part of the composite service. In static approaches, service aggregation takes place at design time when the architecture and the software system design are planned. This type of composition can work properly if the web service environment and the partners involved in the process do not change or rarely change. Static web service composition is not flexible as it is not adaptable to runtime changes, and is too restrictive for inevitable changes in the service requirements. In dynamic approaches, service aggregation occurs at runtime, depending on the customer’s request. It aims to overcome the apparent problems in static web service composition. Dynamic composition techniques are increasingly preferred because of their ability to handle unpredictable changes in the runtime environment that cannot be handled using static techniques. They provide flexibility to modify, extend, and adapt changes at runtime;

- **Manual and Automatic composition**: The second aspect deals with the case where the composition is done manually by a designer, or automatically by a composition system. In manual composition, the composition is obtained by programming via orchestration languages such as WS-BPEL (Web Services Business Process Execution Language). Manual composition is complex and susceptible to errors due to the dynamic behavior and flexibility of the Web. It consumes a lot of time and cannot adapt to the dynamic environment. In automatic web service composition, the system processes the user’s request and automatically composes services to achieve this goal. Automation minimize user intervention, reduces the time spent in order to create a composition plan with respect to time required in a manual composition approach, eliminates human errors and reduces the overall cost of the process.

This article focuses on Automatic Dynamic web service composition.

The web service environment is very dynamic by nature. The number of web service providers is steadily increasing, resulting in a tremendous increase in the scale of web services that use different control structures to exchange data streams. In such a dynamic environment, the web service composition process becomes a tedious task. Furthermore, due to the vulnerable and distributed nature of the Web, a web service can experience failures caused by either server-side software bugs, physical defects in the underlying network infrastructure or failing servers (Chan et al., 2007). In such circumstances, the entire composite service must be fully executed again. A well-designed service composition should take this fact into account. However, ensuring the availability of services seems a very difficult challenge because of the confidentiality and autonomy of each service. Therefore, a recovery mechanism is needed to ensure that the execution of service composition is not interrupted and the failing service is replaced quickly and efficiently.

Several methods and tools have dealt with service composition in recent years. The majority of them have some drawbacks such as scalability and failure problems, high complexity, high computational cost and memory intensive. The graph model is gaining popularity in the community. It overcomes the problem of scalability encountered by other models. Some recent works consider the problem of the composition of service as a graph search problem where a search algorithm is applied over a dependency graph to retrieve service composition results. In these proposals, the failure recovery for reliable execution has received relatively limited attention, the problem of data redundancy is not addressed, which complicates the search process and affects the scalability of the solution. In addition, both the construction and the search of the graph are done at the runtime (Upon receipt of the user query) and this has a negative impact on the execution time.

In this article, a graph-based approach “DynaComp” for optimizing web service composition is proposed. Our approach is based on representing the semantic relationship between web services using
Affordances of Data Science in Agriculture, Manufacturing, and Education

Dynamically Adaptable Web Services Based on the Simple Object Access Protocol
[www.igi-global.com/chapter/dynamically-adaptable-web-services-based/26913?camid=4v1a](www.igi-global.com/chapter/dynamically-adaptable-web-services-based/26913?camid=4v1a)

Web Services Discovery with Rough Sets
[www.igi-global.com/article/web-services-discovery-rough-sets/3134?camid=4v1a](www.igi-global.com/article/web-services-discovery-rough-sets/3134?camid=4v1a)