Chapter 7

Selective Querying for Adapting Hierarchical Web Service Compositions

John Harney
University of Georgia, USA

Prashant Doshi
University of Georgia, USA

ABSTRACT

Web Service compositions (WSC) often operate in volatile environments where the parameters of the component services change during execution. To remain optimal, the WSC could adapt to these changes by querying the participating providers for their revised parameters. Previously, the value of changed information (VOC) has been utilized in simple WSCs to selectively query only those services whose revised parameters are expected to bring about significant changes in the composition. In many cases, however, in order to promote scalability, a WSC is formulated as a more complex, nested structure—a higher-level WSC may be composed of WSs and lower-level WSCs—inducing a natural hierarchy over the composition. This chapter presents a novel approach that extends the capabilities of VOC-driven querying to address the problem of adapting hierarchical WSCs. It shows how to compose and adapt hierarchical WSCs by first deriving a model of volatility for lower-level WSCs and then by descending down the levels of nesting and computing the VOC for WSCs at each level. Experimental results demonstrate that this approach provides an effective and efficient solution for complex, hierarchical WSCs.

INTRODUCTION

Traditional approaches for composing Web Services (WS) assume that the parameters used to model the environment remain static and accurate throughout the composition’s execution lifetime. This is especially true of approaches that use classical planning for composition. Web Service compositions (WSC) are built using a pre-defined,
Selective Querying for Adapting Hierarchical Web Service Compositions

fixed model of the environment at design time, and executed. However, this fundamental assumption is often unrealistic as environments tend to be transient. For example, a product may go out of stock affecting its availability, the network bandwidth may fluctuate affecting the WS response time, or the cost of invoking a travel agent’s service may increase. Many WSC techniques do not adapt compositions to such changes, leading to suboptimality.

Dynamism manifests in WSC environments in a variety of ways. For example, changes range from the operational level (such as a newly introduced task) to the organizational level, such as new company policies (Aalst & Jablonski, 2000; Han & Bussler, 1998). Indeed, these surveys classify a variety of changes in different ways. Solutions have been presented to address some of these changes ranging from exception handling techniques defined in Borgida and Murata (1999) to instituting protocol adaptations (as in Desai, Chopra, & Singh, 2006).

However, less attention has been paid to data volatility that exists during execution. As a concrete example, consider a mortgage loan acquisition process in which two title insurance agencies compete for orders from a large mortgage broker. The sequence in which the broker utilizes the services of the two insurers would depend on the probability with which the insurers usually satisfy the requests and the costs of using them. If the preferred insurer’s rate of request satisfaction drops suddenly (due to say, a financial crisis), a cost-conscious broker should replace it with another insurer to remain optimal. Important non-functional service parameters such as cost, availability, or the rate of request satisfaction in the above example, often change during the lifecycle of a WSC. WSCs must be aware of the changing parameters of the participating services so as to optimize the composition. 1

Thus, the WSC must possess up-to-date knowledge of the revised information during execution. To obtain this knowledge, an adaptive WSC may query services—typically their providers—for the services’ revised parameter values. The revised values are then integrated into the model so that the composition is optimal.

Querying for component services’ parameters, however, comes with its own attendant challenges. While revised information about some services may lead to changes in the WSC, changes to other services’ parameters may have little or no impact on the WSC. Additionally, WSCs typically operate over an open and large scale system (the Web) due to which querying for information from service resources could get tedious, time consuming, and costly. Queries must therefore be carefully managed—we should query those services only whose parameter changes may potentially impact the WSC while minimizing any additional overhead introduced. Specifically, the adaptive WSC should know: (1) when it is cost effective to query for the changed information and, (2) which service(s) to query.

Previously, we introduced a method that guides intelligently query of revised service parameters using the value of changed information (VOC) (Harney & Doshi, 2009a). In particular, we compute the trade-off between the cost of querying for up-to-date information 2 and the value of expected change in the WSC that the revised information is expected to bring. We update the model parameters and compose the WSC again, only if the VOC is greater than the query cost. In computing the VOC, we utilize stochastic models of volatility of each of the services’ parameters. We adopt a myopic approach in that we query only one service provider at a time and utilize the revised information for that WS which leads to the maximum VOC.

We previously demonstrated the usefulness of VOC in the context of simple WSCs (Harney & Doshi, 2009a). In this article, we generalize the applicability of VOC to a hierarchical WSC. To promote scalability a WSC may be often nested—a