Chapter 9

Adaptive Service Binding with Lightweight Semantic Web Services

Carlos Pedrinaci, Dave Lambert, Maria Maleshkova, Dong Liu, John Domingue, and Reto Krummenacher

Abstract Adaptive service selection is acknowledged to provide a certain number of advantages to optimise the service provisioning process or to cater for advanced service brokering. Semantic Web Services, that is services that have been enriched with semantic annotations have often been used for providing adaptive service selection by deferring the binding of services until runtime. Thus far, however, research on Semantic Web Services has mainly been dominated by rich conceptual frameworks such as WSMO and OWL-S which require a significant effort towards the annotation of services and rely on complex reasoning for which there are no efficient solutions that can scale to the Web yet. In this chapter, inline with current trends on the Semantic Web that sacrifice expressivity in favour of performance, we present a novel approach to providing adaptive service selection that relies on simple conceptual models for services and less expressive formalisms for which there currently exist...
mature and performant implementations. In particular, we present a set of conceptual models defined in RDF(S) that support both Web services and Web APIs and we show how simple templates abstracting user requirements can be automatically transformed into SPARQL to enable service selection in a scalable manner.

9.1 Introduction

Web services provide means for encapsulating software functionality as remotely accessible components, independent of programming language and platform. Considerable effort has been devoted to defining architectures, developing communication middleware, and creating languages and process execution engines that can support the creation of complex distributed systems by seamlessly combining Web services. Service-oriented architectures (SOAs) advocate the development of solutions whereby service providers advertise the services they offer in a shared and publicly accessible repository. Software developers or intelligent applications can then access this repository in order to find suitable services for a given purpose and subsequently invoke them.

Web services have increasingly been used within and in some cases between enterprises. However, despite the essential advantages brought by service-oriented technologies, their use in enterprise settings is not without problems. For instance, the execution of business processes defined in this manner typically relies on rigid process models which interact with a fixed and predefined set of partner services. This rigidity impedes or at least complicates to a large extent very desirable features like the dynamic replacement of services based on their current state, the selection of those that better fit a certain context, etc. Conventional solutions to such problems are brute-force: for example, modifying the process models with somewhat artificial branches. This approach results in models that are more complex, and adapting as well as maintaining them in the light of changing conditions turns out to be a hard task [36]. These limitations are even more important in open environments like the Web, where additional difficulties appear such as the heterogeneity of data formats or the unreliability of servers. Recent trends indicate that other technologies, such as HTTP-based Web APIs, are preferred in these cases [11].

In this chapter we address the rigidity of business processes by providing adaptive service selection. This kind of technique, also known as late-binding, relies on deferring the selection and binding to the service to be executed until runtime so that up to date detailed information concerning the state of the business process and other contextual factors such as the previously monitored performance of services can help to adapt the selection to the (presumably) most optimal or appropriate solution. Adaptive service selection has often been based on exploiting semantic annotations of services. Research in this area has thus far been mainly driven by rich Semantic Web Services (SWS) conceptual models such as the Web Service Modeling Ontology (WSMO) [17] and OWL-S [29], which rely on expressive knowledge representation formalisms such as Web Service Modeling Language (WSML) [7] and OWL [35].