ABSTRACT

Web service technology is an open standards-based mechanism for communication over a network. Web services are simple, self-contained applications that perform functions, from simple requests to complex business processes. Client applications can integrate an existing Web service and communicate with it over a network, regardless of the hardware or platform used on either side of the wire. As Web services enable computer-to-computer communications in a heterogeneous environment, they are ideally suited for the Internet. We have observed that the implementation and deployment of a service-oriented application where several third-party Web services need to be integrated, pose a variety of challenge. The main concern of the research presented in this dissertation is the dynamic integration, composition, selection and management of Web services in client applications. The current approaches dealing with these processes fall short in providing the needed runtime flexibility to deal with various events in a dynamic services environment. Web services get hard-wired in the client, or only limited flexibility is provided to integrate functionally equivalent services. There is no explicit support for key requirements such as compositional mismatches, hot-swapping and service compositions. Furthermore, if client applications want to take into account Quality-of-Service properties in order to select the most optimal service for a given request, there is no way to enforce selection policies without explicitly providing code in the client in advance. Finally, we observe that invoking third-party services is far more complicated than making local method invocations. A variety of concerns needs to be dealt with, including exception handling, billing, logging, service monitoring, authentication, etc. All of these concerns need to be reflected in the code of the client. Current approaches require that code dealing with these concerns is provided explicitly and in advance, leading to tangled and scattered code while not being able to deal with unanticipated concerns.

In this dissertation we propose a mediation framework that takes care of all service related concerns for a client application. The framework, called Web Services Management Layer (WSML) introduces a flexible service redirection mechanism that allows for the run-time integration of services and compositions while taking into account selection policies and enforcing a range of client-side management concerns. As a lot of code dealing with the various identified service concerns results crosscutting in the client at those places where a service functionality is required, we opt to employ Aspect-Oriented Programming (AOP) to achieve a better separation of concerns. Service communication and composition details, selection policies and service management concerns are all ideal candidates to be modularized in aspects. As the WSML requires a lot of runtime flexibility, we opt to use a dynamic AOP approach, which allows for the hot deployment of aspects. The WSML offers support during the development, deployment and runtime cycle of service-based applications. To avoid that WSML administrators require aspect-oriented knowledge, the use of aspects is hidden by using aspect libraries and by doing automatic generation of aspect code. A prototype of the WSML has been developed in Java and JAsCo, a dynamic AOP language, and is deployed on a broadband provisioning platform of Alcatel.