Chapter VII

RAWS & UWAS: Reflective and Adaptable Web Services Running on the Universal Web Application Server

Javier Parra-Fuente, Oxford University, UK
Salvador Sánchez-Alonso, University of Alcala, Spain
Marta Fernández-Alarcón, Pontifical University of Salamanca, Spain

Abstract

Reflection is a powerful technology that allows us to produce auto-adaptable software. RAWS is a reflective, multilevel Web service architecture aimed at allowing a Web service to transform its structure and behaviour without the need of human intervention to change the source code, compile it or deploy it again on the application server. Using RAWS, the Web service can change itself automatically. Current application servers have a very important limitation: The deployment platform (J2EE, .NET, etc.). Using current servers, a Web service or application can only be deployed on a server which runs with the same technology. To solve this drawback, we have developed universal Web application server (UWAS), a platform capable
of deploying Web services or applications written in any object-oriented language or for any platform. This is possible thanks to the fact that UWAS internally uses a language-independent object-oriented Web server markup language (OOWSML) representation based on XML. Altogether, RAWS & UWAS make it possible to deploy a Web service on the server regardless of its implementation technology, providing the flexibility to automatically adapt or transform the Web service structure and/or behaviour.

---

Introduction

Web services are programmable components of applications that use SOAP (Gudgin et al., 2003) as an access protocol, regardless of their client and component technology (a drawback in DCOM) and regardless of the language in which both communication ends are written (a drawback in RMI). SOAP generally uses the HTTP transport protocol over the port 80 for request/response, thus crossing corporate firewalls (a drawback in CORBA or DCOM) and facilitating the interoperability of applications that work with different technologies.

Currently, the modification of a Web service implies the availability, edition, recompilation and redeployment of the source code. Depending on the application server, the deployment can either be a simple task or a very complicated work. If the application server supports the dynamic load of applications, then the deployment will be simple task; but if that is not the case, it will imply to stop the execution of the Web service, replacing the old version with the new one, and deploying the new version.

Reflection is a property of computational systems that allows them to reason and act by themselves and to modify their behavior (Maes, 1987). Although this concept has been successfully applied to other fields, such as distributed systems (Ledoux, 1999; McAffer, 1995), concurrent programming (Masuhara, Matsouka, & Yonezawa, 1993), aspect-oriented programming (Pawlak, Duchien, & Florin, 1999; Tanter et. al., 2003), its application to Web services design has not been addressed yet. Reflection can be applied to Web services in order to enhance their adaptability and flexibility. We propose in this chapter a Web service reflective architecture, RAWS, which dynamically modifies a Web service during its execution.

In this chapter, we introduce the basic concepts of reflection that will be applied to Web services (the introspective characteristics and the analysis of the structural and behavioral reflection of the Web service), the architecture model of a reflective and adaptable Web service, and the automatic generation mechanism to obtain the reflective infrastructure needed for a Web service to be dynamically adaptable.
A Dynamic Label Checking Approach for Information Flow Control in Web Services


[www.igi-global.com/article/dynamic-label-checking-approach-information/3072?camid=4v1a](www.igi-global.com/article/dynamic-label-checking-approach-information/3072?camid=4v1a)