

Adapting Web Services for Multiple Devices: a Model-Driven, Aspect-Oriented Approach

Guadalupe Ortiz¹, Alfonso García de Prado

University of Cádiz

C/ Chile 1, 11002 Cádiz, Spain

{guadalupe.ortiz@uca.es, alfonso.garciadeprado@alum.uca.es}

Abstract. Mobile devices have become an essential element in our daily lives, even for connecting to the Internet. Web services have become extremely important when offering services through the Internet. However, current Web Services are very inflexible as regards their invocation from different types of device, especially if we consider the need for them to be adaptable when being invoked from a mobile device. In this paper, we will propose several alternatives for the creation of flexible Web services which can be invoked from different types of device, and compare the different proposed approaches. Aspect-Oriented Programming and Model-Driven Development have been used in all proposals to reduce the impact of service adaption, not only for the service developer, but also to maintain the correct code structure.

Keywords: Web Services, Mobile Devices, Aspect-Oriented Programming, Model-Driven Development.

1 Introduction

Mobile devices have acquired great prominence over the last years. The great amount of devices and its non-stop use gives us a clear picture of the importance of access to mobile services. E-Service developers have focused mainly on the development of services designed to be accessible from desktop computers, creating a void in the sphere of their access from mobile clients – PDAs, mobile phones, etc.-, domain which is becoming a usual scenario. In order to meet this requirement we have to bear in mind the type of device from which the service is going to be invoked.

In this regard, the developed client will vary widely depending on the target device: there may not only be a big difference between a mobile device or a desktop computer, but also between the client developed for different types of mobile devices. In this paper, we explore several solutions for the creation of services which can be invoked from different devices, providing each one with the appropriate response. We also analyze the advantages and drawbacks of each alternative so that developers can choose the option which better adapts to his requirements.

In the following section we explore the three alternatives proposed for services adaptation to mobile devices; further information can be found in the Proceedings of the 2009 Congress on Services - I - Volume 00, Pages 754-761.

¹ The first author was working at the University of Extremadura when she carry out this research.

2 Alternatives for Creating Web Services for Multiple Devices

The first option for the creation of a service which personalizes the answer to received invocations depending on the type of device used is to provide the client with the possibility of adding a new parameter in the invocation. This parameter needs not be provided by the desktop user, but by the mobile one. By default the service will return all the information associated with the request, and only if the customer indicates he has a CDC or CLDC device in the invocation, will we return the relevant information. In regard to the service side, the developer will build the Web service as usual (provided that the types are consistent with mobile ones) and we will make use of aspect-oriented programming to adapt the answer should that be necessary. That is, the aspect, depending on the value of the device parameter, will select which information has to be provided and which does not.

The second option for creating a service which personalizes the answer of the received invocation is adding an optional tag in the invocation SOAP message header in which the client can point out from which type of device he is performing the invocation. It might be expected that in this case we would follow the decision of returning the full information by default, as in the previous subsection. However due to the limited number of APIs to be used with CLDC and to facilitate as much as possible the task of client developers, we decided that the service will return the CLDC information by default and, when indicated in the header, CDC or full information will be provided. For the selection of information, as in the previous case we will make use of an aspect. The developer will develop the Web service as usual and then the aspect will adapt its result depending on the header content.

Our last proposed option which personalizes the answer of received invocations depending on the invoking device is to have different operations depending on the type of device. Of course this would imply some redundant code in the service, however it would not be very significant if we were to make use of the aspects. In this case, the user would invoke the appropriate operation according to the requesting device. All operations invoke a common method which obtains all the book information. The aspect would intercept the invocation to its method and let the query to the database proceed. It would then adapt the results, which have to be provided depending on the device used, and let the result return to the invoker.

3 Conclusions

Web services are not flexible enough to be invoked from different types of client. In this paper we have provided several approaches for the efficient adaptability of Web services. The presented approaches intend to be non-intrusive with services' functionality code, adapting the latter to the device through the use of aspect-oriented programming. Besides, the fact that we followed a model-driven development facilitates the acquisition of these approaches at a low development cost.

Acknowledgments. This work has been developed thanks to the support of MEC (contract TIN2008-02985).