Adapting Web Services for Multiple Devices

Guadalupe Ortiz
Quercus Software Engineering Group
University of Extremadura, Spain
OUTLINE

• INTRODUCTION
• MOTIVATING SCENARIO
• ADAPTING WEB SERVICES TO MOBILE DEVICES
• MDD OF MOBILE-AWARE WEB SERVICES
• EVALUATION
• CONCLUSIONS AND FUTURE WORK
INTRODUCTION

• Mobiles are used 24 hours, seven days a week…
• In the scope of Web 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.

• From the server-side point of view, we consider it necessary that services are built to be invoked from different type of devices, returning them the appropriate response depending on the invoking device.

• This should be done in a modularized and decoupled way, without creating intrusive code in the service main functionality code.
MOTIVATING SCENARIO

• We have an e-commerce service such as an online bookstore.
• We have all the information about a specific book available in the website.
• We employ a new staff member to visit prospective customers and we provide him with a PDA.
• The new staff member does not need to see the opinions of other users about the book.
• The customer is interested in a book and is out for lunch and decides to buy the book through his mobile phone…
• … Three different clients for the same service
ADAPTING WEB SERVICES TO MOBILE DEVICES

Alternatives to the Creation of Device-Aware Web Services

– Including a new tag in the SOAP message header.

– Including a new parameter in the invocation.

– Offering different operations for different services.

– Adapting the results making use of façades.
ADAPTING WEB SERVICES TO MOBILE DEVICES
Implementation Details for the Creation of Device-Aware Web Services (i)

getBookInfo(ISBN)
SOAP header: device="laptop"

BOOKSTORE WEB SERVICE

getBookInfo(ISBN)
SOAP header: device="cdlc"
ADAPTING WEB SERVICES TO MOBILE DEVICES
Implementation Details for the Creation of Device-Aware Web Services (ii)

Client

Web Service Interface

Device-Related Aspect

Soap Handler

getInfo(ISBN) -> Proceed(ISBN) -> return result -> return adapted result

getInfoHeader
• This gives the developer the option of building Web services for a current computer invoker and then extend them for mobile users should that be necessary –through the use of aspect-oriented techniques-, without modifying the original code or adding any intrusive one.
ADAPTING WEB SERVICES TO MOBILE DEVICES

Expected Results

Title: Web Services: Principles and Technology

Author: Michael Papazoglou

ISBN: 321155555

Web services, usually including some combination of programming and data, are made available from a business’s web server for web users and other web connected programs. The accelerating creation and availability of these services is a major computing trend as software becomes increasingly distributed and web-based. Web services are the next logical step for web-based computing and will have a profound impact on the way in which business is conducted on the web in the future. As they involve many different systems communicating with each other, they are particularly important following the proliferation of the range of computing devices (PDA’s, mobile telephones, hand held computers etc.). This book will provide a comprehensive treatment of the concepts and uses in web services, looking at how they are designed, and the key technologies, and standards used.

Publisher: Prentice Hall

Price: 67.50 Euros

Pages: 784

Comments: This book is an excellent and authoritative journey into the world of Web Services. It is an incredible read! Highly recommended!!
MDD OF MOBILE-AWARE WEB SERVICES

• Once we know which is the best alternative for the implementation of mobile-aware Web services and what their implementation requirements are, we propose a model-driven aspect-oriented development approach.

• We will have
  – A platform-independent model
  – A platform-specific model
  – Transformation from PIM to PSM
  – Automatic generation of AspectJ code from PSM
MDD OF MOBILE-AWARE WEB SERVICES

Platform-Independent Model
MDD OF MOBILE-AWARE WEB SERVICES
Platform-Specific Model
MDD OF MOBILE-AWARE WEB SERVICES
PIM-PSM and PSM-Code Transformation Rules

PIM-PSM:
• We have developed a plugin for Eclipse using Model to Model Transformation Authoring. The plugin consists of the definition of a set of transformations from a UML model into another UML model.
• Once transformation rules have been defined, their Java code is generated automatically and the plugin is ready to be executed as an Eclipse application.

PSM-CODE
• We assume that there are several platforms and tools which can generate code for Web services from their offered interface as well as for complex type definition classes.
• For generating the aspect code to adapt the service to the different invoking types we propose developing transformations through the use of Eclipse Project M2T (Model to Text) and specifically of Jet Templates with Exemplars.
EVALUATION
Aspect-Oriented Evaluation and Use of Headers and Handlers Evaluation

Average performance time per invocation

Time in ms

100 1000 50000 500000
Java 18.07 17.61 17.52 17.40
AspectJ 17.51 17.52 17.47 17.39
EVALUATION

Additional Issues

• In regards with the model-driven development, the learning curve due to the incorporation of AOP code is avoided for the service developer, as well as saving the added workload of making services device-aware.

• At the same time, we prevent possible coding errors and facilitate quick maintenance of aspect-oriented classes by providing new transformation files, when required.
CONCLUSIONS AND FUTURE WORK (i)

- The approach presented provides us with the possibility of following a model-driven development of mobile-aware Web services in an integrated platform.
- Besides, the approach is perfectly extensible to different types of result the service has to return (not only mobile phone versus computer).
- We are also dealing with adaptation of the client side to specific models and user preferences (not relevant in this scope)

Future Work… (Suggestions are welcomed!)
- A proper definition of the Computational-Independent Model and its automated processing for the inclusion of stereotypes in the PIM model would no doubt enrich the presented approach and is planned for our future research
- Additional Evaluation: model checking …