Chapter 5
Towards Event-Driven Context-Aware Web Services

Guadalupe Ortiz
Quercus Software Engineering Group, Spain &
UCASE Software Engineering Group, Spain

Juan Boubeta-Puig
UCASE Software Engineering Group, Spain

Alfonso García de Prado
UCASE Software Engineering Group, Spain

Inmaculada Medina-Bulo
UCASE Software Engineering Group, Spain

ABSTRACT
Web services provide a successful way to communicate distributed applications, in a platform-independent and loosely coupled manner. Even though there are examples of good practice for the design, development, and management of web services, getting services to be context-aware is still under investigation. Current proposals require communication with an external context server or manager, slowing down service performance. In this work, the authors propose an architecture which utilizes complex event processing for detecting context events relevant to the services in question and an aspect-oriented adaptation, maintaining a loosely coupled service implementation as well as keeping its main functionality structure without adding any context-related intrusive code.

INTRODUCTION
In recent years, Service-Oriented Architectures (SOAs) have emerged as an efficient solution for the implementation of systems in which modularity and communication among third parties are key factors. This fact has led to the increasing development of distributed applications composed of reusable and sharable components (services). These components have well-defined platform-independent interfaces, which allow SOA-based systems to quickly and easily adapt to changing business conditions.

However, although there are good procedures for the design, development and management of Web services, there are scopes where Web service
adaptation is required. For instance, we may have services that would be suitable for the adaptation of the invoking client’s specific context, service answers which should be adapted depending on the type of invoking device or services which are to return different answers depending on the specific client using it. In the past, we proposed a method for adapting services to the invoking device (Ortiz & Garcia de Prado, 2010a) as well as to adapt them to the client-specific context (Ortiz, & Garcia de Prado, 2010b); in this work, we will tackle their adaptation to the external context, which should be implemented appropriately, so that the loosely-coupled nature of web services is maintained. Previous approaches are good for the specific type of context dealt with –adapting to device and client-specific context– but are not prepared to deal with the external context, which requires of additional mechanisms in order to be able to capture the latter as well as detect what is relevant for the current service invocation.

In this regard, adapting services to context and current conditions might require the analysis of context information very often. Nevertheless, service-oriented architectures are not suitable for environments where it is necessary to continuously analyze the information flowing through the system, a key factor for an appropriate context-aware service implementation. This limitation may be solved by the joint use of Complex Event Processing (CEP) (Luckham, 2002) together with SOA. CEP can process and analyze large amounts of events and correlate them to detect critical or relevant information; in this scope event patterns are used to infer new more complex and meaningful events related to context.

However, most approaches implementing context-aware services do not use CEP or are not using it for this purpose, therefore having to continuously access a context server or manager.

Besides, requirements for service adaptation to context might change along the useful life of a service; since new relevant contexts might be taken into account or some older ones might be discarded because they are not considered relevant any more. This leads to the fact that we need an approach which provides service adaptation in a completely decoupled way to facilitate service maintenance and evolution. In our previous approaches, Aspect-Oriented Programming (AOP) demonstrated to be a good alternative to decouple context adaptation from service implementation.

As a result of the facts described in previous paragraphs, in this work, we propose the use of CEP in conjunction with aspect-oriented adapters, to avoid services having to access external modules for their adaptation to context information and to ease their maintenance and evolution.

The rest of the chapter is organized as follows: Section 2 provides a short background on context-awareness, complex event processing and aspect-oriented programming. Then, Section 3 explains the envisaged architecture as well as provides an illustrating scenario to facilitate the proposal understanding. Afterwards, Section 4 outlines main related work. Finally, conclusions and future work are provided in Section 5.

BACKGROUND

In this section we will introduce the concept of context-awareness, complex event processing and aspect-oriented programming.

Context-Awareness

Dey et al.’s context definition in (Abowd et al., 1999) is specially well-known –page 3, section 2.2: “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”.

The term context-awareness supports the fact that the context information provided by the client, or taken from the environment, is properly used