# Chapter 15
# A Middleware Architecture for Developing Mobile Applications

**Hana Rubinsztejn**
*Federal University of Mato Grosso do Sul, Brazil*

**Ricardo Rocha**
*Institute of Informatics, Federal University of Goias, Brazil*

**José Viterbo**
*Federal Fluminense University, Brazil*

**Gustavo Baptista**
*Pontifical Catholic University of Rio de Janeiro, Brazil*

**Vagner Sacramento**
*Institute of Informatics, Federal University of Goias, Brazil*

**Markus Endler**
*Pontifical Catholic University of Rio de Janeiro, Brazil*

## ABSTRACT

*In this chapter, the authors present MoCA, a service-oriented middleware architecture that supports the development and deployment of distributed applications for mobile devices, which are inherently context-aware and adaptive. Besides explaining its main services and APIs, they discuss in which ways the MoCA architecture supports some well-known software engineering principles that apply to the design and implementation of context- aware applications. Furthermore, the authors give an overview of its usage and present prototype applications that have been developed on the top of MoCA.*

## INTRODUCTION

The development of distributed applications that run on mobile devices, which have to opportunistically adapt to the conditions of their environments, may be greatly simplified by the use of context-provisioning middleware systems (Schilit, Adams, & Want, 1994). Nevertheless, despite the huge amount of publications describing elaborated context models (Strang, & Linnhoff-Popien, 2004) and extensible frameworks or middleware systems (Baldauf, Dustdar, & Rosenberg, 2007), unfortunately, to date there are only very few freely available, and, in fact, easily usable systems for the development of context-aware mobile applications.

In this chapter, we report our experience in effectively using the *Mobile Collaboration Architecture* (MoCA) (Sacramento et al., 2004), a freely available and easily usable service-based

DOI: 10.4018/978-1-4666-4301-7.ch015

architecture that offers support for the development of distributed context-aware applications for mobile devices interconnected through IEEE 802.11 wireless LANs. MoCA's services and components provide means of collecting, distributing and processing context data obtained directly from the mobile devices, i.e., the state of the devices' resources, as well as, parameters of the wireless network connection. MoCA architecture is extensible to allow the incremental development of additional context producing and processing services. In addition, it makes available to the application developer a set of APIs for synchronous and asynchronous access to context data and other context-specific services.

Presently, MoCA is a stable service-oriented middleware architecture that has been effectively used by several research groups in Brazil and abroad for the development of small-scale or experimental context aware applications. Statements and evaluation grades given by some developers to several aspects of the architecture such as ease of installation and use, online documentation, robustness and reliability show that MoCA succeeded in meeting the developer's needs.

In the next section we discuss the goals and requirements that guided the MoCA's design and brie⟩y discuss how it meets some well-known principles of system's software engineering, facilitating the development of context-aware mobile applications. Next, we present the MoCA's architecture, its basic services and the set of context data that it delivers, some optional services and MoCA's personalities and extensions. Then, we present the main APIs and the typical use for context-awareness in MoCA-based applications. After that, we present a few selected context-aware application prototypes that have been developed over the years using MoCA. Then, we compare MoCA with other context provisioning middleware platforms commonly referenced in literature. Finally, last section draws some conclusions and points to future developments.

## THE DESIGN OF MoCA

It has been well recognized that the development of context-aware and adaptive mobile applications is a complex task and requires the careful observance of several well-known software engineering principles (Ghezzi, Jazayeri, & Mandrioli,1991; Roman, Picco, & Murphy, 2000). Some of these principles guided our design and development of the MoCA architecture itself and, therefore, were fundamental to the success of the project. As shown in Figure 1, MoCA was designed as a layered architecture following a context server approach (Baldauf, et al., 2007), in which largely independent services provide an infrastructure for collecting, distributing and processing context information. This approach aimed at facilitating the development of applications by observing principles such as separation of concerns, multi-level abstractions, incremental development, flexibility of customization and multi-language and interoperability support.

*Separation of concerns* is absolutely necessary to cope with the complexity of designing distributed mobile applications. MoCA addresses this question by making context access transparent to the application developer. It provides a coherent and intuitive interface to access system's context and location information. *Abstractions* are the natural means by which separation of concerns is realized by the middleware, hiding the low-level resources but still making explicit the key concepts involved in the development of mobile applications, and as such allowing application developers to have the appropriate level of knowledge about the computational scenario (Roman et al., 2000). MoCA provides different levels of abstraction, ranging from the complex notion of a hierarchy of *symbolic regions* down to the simple context value change event. While MoCA's independent services permit separation of concerns, the simple and comprehensive set of APIs available offer multi-level abstractions for the application devel-