Chapter 3.7
DSOA:
A Service Oriented Architecture for Ubiquitous Applications

Fabricio Nogueira Buzeto
Universidade de Brasília, Brazil

Carlos Botelho de Paula Filho
Universidade de Brasília, Brazil

Carla Denise Castanho
Universidade de Brasília, Brazil

Ricardo Pezzuol Jacobi
Universidade de Brasília, Brazil

ABSTRACT

Ubiquitous environments are composed by a wide variety of devices, each one with different characteristics like communication protocol, programming and hardware platforms. These devices range from powerful equipment, like PCs, to limited ones, like cell phones, sensors, and actuators. The services provided by a ubiquitous environment rely on the interaction among devices. In order to support the development of applications in this context, the heterogeneity of communication protocols must be abstracted and the functionalities dynamically provided by devices should be easily available to application developers. This paper proposes a Device Service Oriented Architecture (DSOA) as an abstraction layer to help organize devices and its resources in a ubiquitous environment, while hiding details about communication protocols from developers. Based on DSOA, a lightweight middleware (uOS) and a high level protocol (uP) were developed. A use case is presented to illustrate the application of these concepts.

1. INTRODUCTION

The presence of computer devices endowed with processing power and communication capabilities grows every day. Taking advantage of these capabilities in order to assist the user in its tasks demanding as minimum attention as possible is the basis of ubiquitous computing (Weiser & Brow, 1995). A smart space (Weiser, 1993) is an environment where the resources made available by the devices are organized and coordinated in order to provide intelligent services to the users.

DOI: 10.4018/978-1-61350-456-7.ch3.7

Such intelligence is implemented in the smart space through applications. In order to abstract and simplify the complexity for building these applications, it is common place the adoption of middlewares that facilitate the implementation of smart spaces. We can highlight some important requirements (Abowd, Atkeson, & Essa, 1998) that must be addressed by middlewares (Bernstein, 1996) for ubicomp.

- Limited devices are part of the smart space, so its limitations like CPU, memory, bandwidth and battery life must be considered. Such issues can be seen in middlewares like the MundoCore (Aitenbichler, Kangasharju, & Mühlhäuser, 2007).
- Interaction details on how applications and resources cooperate must be addressed. Not only synchronous but asynchronous communication must be taken into account. An example of a middleware in the literature that focus on these issues is MoCA (Sacramento et al., 2005). Communication can occur not only in small messages, but also can be streamlined as in the MediaBroker project (Modahl, Bagrak, Wolenetz, Hutto, & Ramachandran, 2004).
- Platform heterogeneity among devices. The hardware and software must be considered in order to allow the integration of as many devices as possible to the smart space. This type of concern is observed in projects like the Mundo-Core (Aitenbichler et al., 2007) and WSAMI (Issarny, Sacchetti, Chibout, Dalouche, & Musolesi, 2005) which provide solutions to multiple programming platforms.

Many middleware solutions, like the WSAMI project (Issarny et al., 2005) and the Home-SOA (The OSGi Alliance, 2009) utilize a SOA (MacKenzie, 2006) based solution. The use of SOA assists in the abstraction of the smart space functionalities in order to simplify the development of smart applications. One problem with a pure SOA approach lies in the fact that it does not address some specific problems of ubicomp environments. For example, SOA does not define the way services are accessed, which is an important aspect in the ubicomp context.

This work proposes an extension of the SOA architecture, denominated DSOA (Device Service Oriented Architecture), to model a smart space taking into account the requirements outlined above. Founded on DSOA, we also developed a lightweight multi-platform communication interface, called uP, and the middle-ware uOS to support the development of smart applications. To illustrate some characteristics of the proposed model a use case, named Hydra Application, is also shown.

This paper is organized as follows. Some related works are presented in Section 2 while Section 3 describes the DSOA architecture. The protocol uP and the middleware uOS are presented in Sections 4 and 5 respectively. Section 6 brings into focus the Hydra Application. Some results and final considerations are addressed in Sections 7 and 8.

2. RELATED WORK

A ubicomp environment is permeated by applications that may run in a variety of devices, including static and mobile ones. In order to easy the development of this kind of software many initiatives can be found in the literature (Garlan, 2002; Cerqueira, 2001; Helal et al., 2005; Brumitt, Meyers, Krumm, Kern, & Shafer, 2000). Among these, we highlight four of them which achieve relevant contribution towards the three requisites presented on section 1.

2.1. MundoCore

The MundoCore (Aitenbichler et al., 2007) middleware focuses on the development of applica-

Related Content

Software-Defined Networking in Aviation: Prospects, Effectiveness, Challenges
Roman Odarchenko (2019). *Cases on Modern Computer Systems in Aviation (pp. 147-175).*
www.irma-international.org/chapter/software-defined-networking-in-aviation/222187

Automatic Static Software Testing Technology for Railway Signaling System
Jong-Gyu Hwang and Hyun-Jeong Jo (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming (pp. 612-630).*
www.irma-international.org/chapter/automatic-static-software-testing-technology-for-railway-signaling-system/261046

Exceptions in Ontologies: A Theoretical Model for Deducing Properties from Topological Axioms
Christophe Jouis, Julien Bourdaillet, Bassel Habib and Jean-Gabriel Ganascia (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications (pp. 61-81).*
www.irma-international.org/chapter/exceptions-ontologies-theoretical-model-deducing/62435

Geospatial Technology: Curricular Keystone of Applied Geography
Richard G. Boehm and Audrey Mohan (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications (pp. 139-148).*
www.irma-international.org/chapter/geospatial-technology-curricular-keystone-applied/62439

Soil Cation Exchange Capacity Predicted by Learning From Multiple Modelling: Forming Multiple Models Run by SVM to Learn From ANN and Its Hybrid With Firefly Algorithm
Rahman Khatibi, Mohammad Ali Ghorbani, Rasoul Jani and Moslem Servati (2018). *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering (pp. 465-480).*
www.irma-international.org/chapter/soil-cation-exchange-capacity-predicted-by-learning-from-multiple-modelling/206762