Computer Standards & Interfaces 36 (2014) 928–940

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Software reference architecture for smart environments: Perception

A. Fernández-Montes [a,*], J.A. Ortega [a], J.I. Sánchez-Venzalá [a], L. González-Abril [b]

[a] ETS Ing. Informática, Universidad de Sevilla, Spain
[b] EU Estudios Empresariales, Universidad de Sevilla, Spain

ARTICLE INFO

Article history:
Received 27 May 2012
Received in revised form 14 January 2014
Accepted 3 February 2014
Available online 19 February 2014

Keywords:
Smart environment
Software architecture
Ambient intelligence
Perception

ABSTRACT

With the increase of intelligent devices, ubiquitous computing is spreading to all scopes of people life. Smart home (or industrial) environments include automation and control devices to save energy, perform tasks, assist and give comfort in order to satisfy specific preferences.

This paper focuses on the proposal for Software Reference Architecture for the development of smart applications and their deployment in smart environments. The motivation for this Reference Architecture and its benefits are also explained. The proposal considers three main processes in the software architecture of these applications: perception, reasoning and acting.

This paper centres attention on the definition of the *Perception* process and provides an example for its implementation and subsequent validation of the proposal.

The software presented implements the *Perception* process of a smart environment for a standard office, by retrieving data from the real world and storing it for further reasoning and acting processes. The objectives of this solution include the provision of comfort for the users and the saving of energy in lighting. Through this verification, it is also shown that developments under this proposal produce major benefits within the software life cycle.

1. Introduction

A smart environment (SE) can be defined as *one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment* [1].

Smart home technologies are an important part of ubiquitous computing. Mark Weiser [2] outlined the principles of Ubiquitous Computing: the purpose of a computer is to help someone do something. Nowadays, due to the popularisation of computational devices and applications, ubiquitous computing is recognised as a revolution in the development of smart environments.

Nevertheless, software artefacts related to ubiquitous computing, together with the wide spectrum of computational devices (and the software needed to fulfil their missions) are too heterogeneous and hence difficult to compare or classify. Each piece of software evolves in an isolated way or only in relation to the hardware for which it has been developed. The problem addressed in this paper involves the orchestration of the architecture of a general software model for the development of SEs.

This Software Reference Architecture would favour the development of a smart environment solution by increasing the reuse of components, promoting interoperability, and defining the competences of each part of the software.

A good comparison for this could be the Open System Interconnection (OSI) model, which is a prescription for characterizing and standardizing the functions of a communications system in terms of abstraction layers.

The ambitious goal of this architecture forces it to remain very general and to leave specific aspects until the implementation stage.

The benefits of the approach include a better understanding of the issues that must be faced when developing each component of a smart environment solution. The Software Reference Architecture reduces the costs of the main cycles of software (design, development, deployment, and maintenance) and favours the interoperability between various solutions.

The main goal of this work is the proposal of Software Reference Architecture for the development of SEs (see Section 3), where all the components can interact flawlessly and reach automatism objectives.

To this end, the architecture proposed seeks to improve the modularity, reusability and extensibility of solutions, thereby allowing a more coordinated evolution of SEs, which currently remain under individual and isolated development. The architecture defines a middleware framework that connects the modules and establishes the responsibility of each module. The benefits for developers using a defined framework or standard architecture for the domain have been thoroughly studied by Fayad and Schmidt [28], and include: a reduction and focus of the effort involved, a soft learning curve, integrability, maintainability, easier validation, efficiency, and a higher level of standardization.

* Corresponding author.
E-mail addresses: afdez@us.es (A. Fernández-Montes), jortega@us.es (J.A. Ortega), jisanchez@us.es (J.I. Sánchez-Venzalá), luisgon@us.es (L. González-Abril).

http://dx.doi.org/10.1016/j.csi.2014.02.004
0920-5489/

As an example of the architecture usage, this paper presents the Perception process and provides an example of implementation by following the Software Reference Architecture proposed.

Typical components of a SE have been thoroughly studied in the literature, although the approach of Cook and Das [20] deserves special mention since it is currently the most widely accepted approach. Fig. 1 shows the general organization of these components. Components are divided into four layers: a) physical; b) communication; c) information; and d) decision. This approach joins hardware with software agents, and hence very heterogeneous elements, such as a decision maker and sensors or actuators, appear in the same component model.

All these components must collaborate in order to achieve the goals of automatism that a SE requires. Which tasks belong to each component and how they should collaborate constitute the main motivation of the Software Reference Architecture proposed.

The Software Reference Architecture proposed is divided into three main parts: Perception, Reasoning and Acting. This paper focuses on the definition of Perception, as the first step in the general process. Section 2 analyses related work in this area, and in Section 3, Reference Architecture is presented and the Perception process is explained. Finally, verification with a prototype of the Perception process is shown in Section 4, and conclusions are drawn in Section 5.

2. Related work

Ambient intelligence is a trending topic, and hence a wide variety of related research initiatives have appeared. One of the most common applications in ambient intelligence is that of SEs. Many researchers around the world are developing projects which involve SEs.

In this section, some of the most popular projects are reviewed and compared. The section has been divided depending on where each project is focused: general smart environments, technologies or architectures.

2.1. Smart environment projects

Da Costa [12] focuses on the challenges and issues that ubiquitous computing applications have to deal with and summarizes them: heterogeneity, scalability, dependability and security, privacy and trust, spontaneous interoperation, mobility, context awareness, context

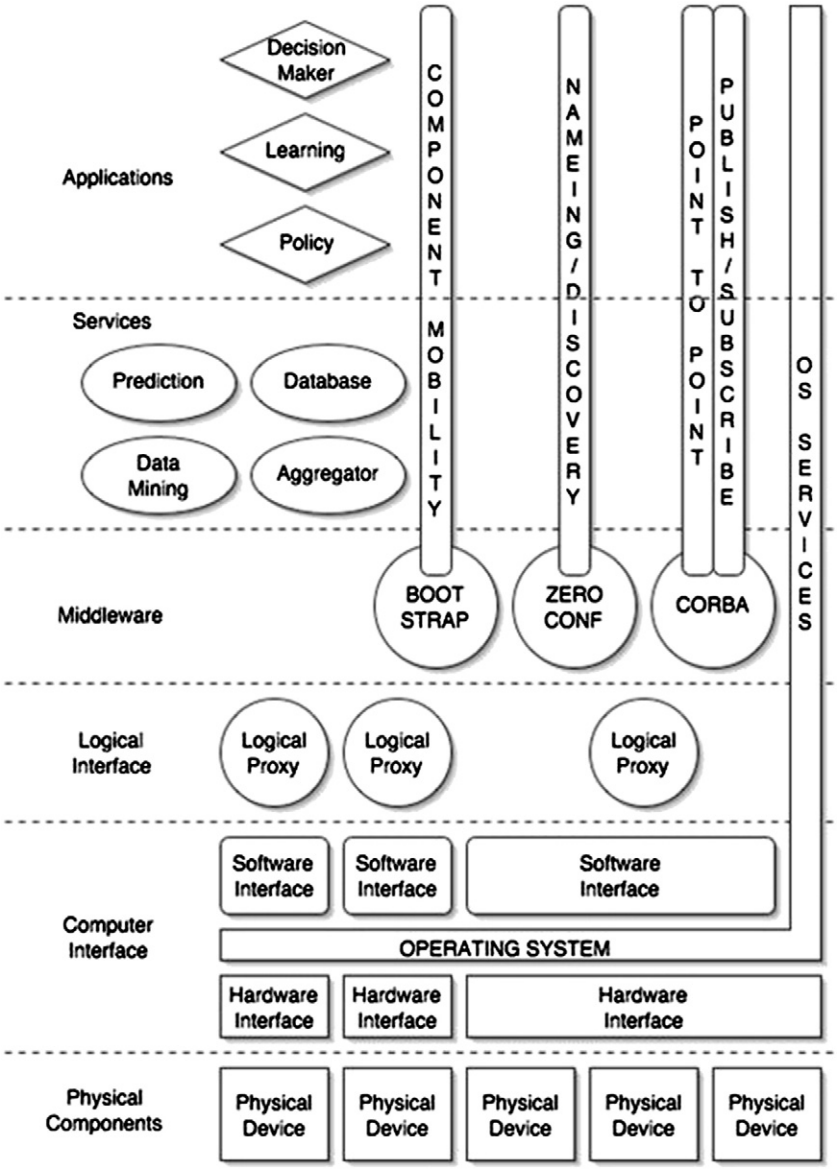

Fig. 1. The components of a SE by Cook and Das [20].