Pervasive and Mobile Computing 8 (2012) 448–466

Contents lists available at SciVerse ScienceDirect

Pervasive and Mobile Computing

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

# SENSOR9K: A testbed for designing and experimenting with WSN-based ambient intelligence applications

Alessandra De Paola, Salvatore Gaglio, Giuseppe Lo Re *, Marco Ortolani

*Department of Computer Engineering, University of Palermo, Viale delle Scienze, ed. 6, 90128 Palermo, Italy*

ARTICLE INFO

*Article history:*
Received 9 August 2010
Received in revised form 11 February 2011
Accepted 20 February 2011
Available online 1 March 2011

*Keywords:*
Wireless sensor and actuator networks
Ambient intelligence
Probabilistic reasoning
Energy efficiency

ABSTRACT

Ambient Intelligence systems are typically characterized by the use of pervasive equipment for monitoring and modifying the environment according to users' needs, and to globally defined constraints.

Our work describes the implementation of a testbed providing the hardware and software tools for the development and management of AmI applications based on wireless sensor and actuator networks, whose main goal is energy saving for global sustainability. A sample application is presented that addresses temperature control in a work environment, through a multi-objective fuzzy controller taking into account users' preferences and energy consumption.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction and motivations

The main goal of Ambient Intelligence (AmI) is the development of systems aimed at adapting the surrounding environmental conditions so that they can match the users' needs, whether those are consciously expressed or not, while at the same time satisfying other system-driven goals, such as the minimization of global energy consumption. An implicit requirement is the use of pervasively deployed sensing and actuating devices, following the ubiquitous computing paradigm which states that technology must not intrude into human lives; hence, control and monitoring devices should be deployed so as to remain invisible to the users [1,2]. Wireless Sensor Networks (WSNs) fully meet these requirements, thanks to their intrinsic pervasiveness and low intrusiveness [3–5], and may thus represent a suitable choice for the sensory layer of AmI systems.

This work presents SENSOR9K, a testbed for designing and experimenting with WSN-based Ambient Intelligence applications, deployed at our labs in the context of our research projects [6]. The name of the testbed is meant to emphasize its pervasiveness, as it ideally recalls the fictional *HAL* 9000 AI system, whose extremities pervaded the spaceship in "2001: A Space Odyssey"; in particular, we intend to address the issue of implementing effective policies for energy saving in the context of indoor environments. Recent studies have shown that ICT technologies, and AmI systems in particular, may play a twofold role since their constituting elements are both significant consumers, and potential actors in steering a more clever overall usage of the available energy resources [7,8]. The pervasive sensory infrastructure may be profitably used to gather information about the current energy usage, as well as the corresponding environmental conditions, into a centralized server, where artificial reasoning techniques may be implemented. Centralizing the reasoning activity preserves its consistency and unitarity [9], and allows one to steer the behavior of the distributed actuators in order to bring the environment to the desired state.

In this context, our testbed aims to boost the development of AmI applications, and we argue that providing an abstraction towards the physical layer by means of a composition of core services will effectively let the AmI designer focus on

* Corresponding author.
*E-mail addresses:* depaola@unipa.it (A. De Paola), gaglio@unipa.it (S. Gaglio), lore@unipa.it (G. Lo Re), ortolani@unipa.it (M. Ortolani).

1574-1192/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2011.02.006

higher-level issues; in this perspective, Sensor9k provides a set of "building blocks" that implement basic intelligent functionalities on top of the underlying distributed sensory and actuating infrastructure.

Sensor9k specifically focuses on indoor environments, where relevant environmental quantities will be monitored through pervasively deployed WSNs without adversely impacting the integrity of the pre-existing structures. Sensor nodes host the software implementing the logic of the application to be tested, as defined by the AmI designer, as well as additional testbed-specific functionalities.

The design of our testbed thus encompasses both hardware and software issues. Besides the pervasive sensory devices, Sensor9k provides a minimal set of communication and processing devices, organized into a backbone of local gateways providing access to the remote WSNs; such intermediate infrastructure is designed in a hierarchical fashion in order to accommodate scalability and fault tolerance, and its main purpose is to act as a connection interface bridging the gap between the distributed sensors and the centralized AmI server. Sensor9k's core is thus represented by a middleware, partly distributed on the remote sensory devices and on the backbone gateways, and partly residing on the central AmI server, which stores the library of modules implementing basic AmI functionalities, such as user profiling, energy monitoring and multi-sensor data fusion, which we regard as the common ground for the creation of applications targeting energy saving.

In order to assess the validity of the proposed testbed, we will present a fully featured sample application, addressing temperature control in the context of a work environment, and involving conflicting goals, namely the satisfaction of the users' preferences in terms of pleasantness of the office environmental conditions, while minimizing the global energy consumption. It will be shown how Sensor9k eases the development of such applications; in particular a multi-objective fuzzy controller will be created by exploiting the basic Sensor9k's functionalities.

The paper is organized as follows: Section 2 summarizes relevant works about software architectures for WSNs in the context of AmI. An overview of the architecture of our system is reported in Section 3, while Sections 4 and 5 contain the details about its physical and middleware layers, respectively. Finally, Section 6 describes our sample application, and Section 7 presents some experimental results.

## 2. Related work

Since their introduction, Wireless Sensor Networks have steadily evolved, especially with respect to the degree of complexity of the network configuration, as summarized in [10]. The point of view has shifted from the use of one single WSN for the entire field, possibly composed of a very large number of nodes, towards a more structured approach involving several interconnected WSNs, each with a limited number of nodes, and up to a comprehensive strategy where the sensor nodes are enabled to interact with diverse devices and applications. Such progress has consequently widened and diversified the range of issues that the different middleware systems for WSNs, presented in literature, attempted to address.

While initial efforts were mainly focused on the optimization of the resources available to the nodes, in terms of energy or computational power, later research has also addressed the functionalities for enabling interoperability among heterogeneous devices and for providing a common interface to different applications. A survey of middleware tools for WSNs must thus consider such variety in the approaches, despite the fact that they may not always be directly comparable.

A traditional categorization of WSN middleware [11,12], mostly focused on the first type of proposals of single WSN deployments, specifically distinguishes them with respect to the adopted programming model; the common goal is always the provision of an intermediate layer decoupling the node application logic from the underlying operating system and hardware. The authors of Impala [13], for instance, adopt a modular design paradigm in order to improve the applications' adaptability, and to provide a simple way to keep them up-to-date; in the authors' vision, the possibility of adapting the application interface on the fly is bound to improve the performance, energy-efficiency, and reliability of the overall system. The modular approach has also led to the adoption of an agent-based programming model, as for instance in [14]; the latter work, in particular, employs mobile agents traversing several nodes, and carrying snippets of code with them; in this view, nodes are able, for instance, to host multiple applications at the same time. Another popular approach involves the use of virtual machines in order to provide the user with programming primitives in an assembly-like language, thus allowing developers to dynamically upload new "scripts" onto the network nodes; systems implementing this approach include Maté [15], and Magnet [16]. Alternatively, the entire WSN has also been viewed as a single distributed database, so that the goal becomes to transparently provide access to sensed data according to the traditional relational model; one of the most relevant works in this context is TinyDB [17].

A limitation of such classification is that it just focuses on middleware running entirely *inside* the network, thus disregarding middleware that instead runs, partly or entirely, *on top* of the sensor network level. The specific context of Ambient Intelligence, on the other hand, has often stimulated researchers to exploit WSNs as a distributed sensory tool, and as a communication infrastructure, whereas the core of the intelligent services typically resides elsewhere, at a higher abstraction level. Our standpoint thus comprehensively considers that most of the AmI system is superimposed over the sensory equipment, and not entirely merged therein; the connection with the sensor nodes is provided by a minimal abstraction layer acting as a wrapper over the (user-defined) application logic, with the aim to hide possible future upgrades to the underlying hardware, as well as to provide a common interface towards the centralized Ambient Intelligence services.

A profitable comparison with other related approaches must thus analyze the functional aspects, besides the architectural paradigm; more specifically, a meaningful distinction for the classification of middleware in this perspective needs to

Download English Version:

https://daneshyari.com/en/article/466126

Download Persian Version:

https://daneshyari.com/article/466126

Daneshyari.com