Ad Hoc Networks 9 (2011) 430–452

Contents lists available at ScienceDirect

Ad Hoc Networks

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

A service-oriented approach to facilitate WSAN application development

Eduardo Cañete, Jaime Chen *, Manuel Díaz, Luis Llopis, Bartolomé Rubio

University of Málaga, Dpto. Lenguajes y ciencias de la Computación, Calle de Bulevard Louis Pasteur, s/n 29071 Málaga, Spain

ARTICLE INFO

Article history:
Received 1 February 2010
Received in revised form 18 August 2010
Accepted 29 August 2010
Available online 8 September 2010

Keywords:
Middleware
Wireless Sensor and Actor Network
Service Oriented Architecture
High-level programming
Framework

ABSTRACT

Due to the complex nature of developing Wireless Sensor and Actor Network (WSAN) applications it is obvious that new frameworks, tools, middleware and higher-level abstractions are needed to make the task of the developers easier. Depending on the WSAN system we want to develop, different characteristics must be taken into account but, perhaps, some of the most important are the capacity to add real-time constraints, the QoS and, of course energy saving. Our proposal USEME is a service-oriented and component-based framework which allows the easy combination of macro-programming and node-centric programming to develop real-time and efficient applications over WSANs. USEME allows the specification of real-time constraints between services, permits the use of groups to structure the network and is platform independent. Two prototypes (Imote2.Net and SunSPOT) have been implemented and several performance tests have been carried out.

1. Introduction

A new world is emerging in which everything—people, mobile phones, cars, etc.—are connected. All of this is called *The Internet of Things*. One of the most attractive and important parts of this kind of Internet is known as the *Wireless Sensor and Actor Networks* [1]. WSANs constitute a new pervasive and ubiquitous technology and are currently one of the most interesting fields of research. Due to a combination of recent technological advances in electronics, nanotechnology, wireless communications, computing, networking, and robotics, it is now possible to design advanced sensors (tiny, low-cost and low-power nodes, colloquially referred to as "motes"), which can be deployed in the environment in order to gather information about physical phenomena and report it to actor devices which are able to react by altering the environment in order to tackle the problem. WSANs offer numerous advantages over traditional systems, for example, they make use of a large-scale flexible architecture (potentially hundreds or thousands of motes), high-resolution sensed data and application adaptive mechanisms. These unique characteristics make WSANs very useful for a wide range of application areas (medicine, security, critical infrastructures, intensive farming, etc.) and as a result the investment being made by industry, university and government organizations is considerable. A recent report from market research firm ONWorld predicts that the global market for WSANs will have grown tenfold by 2011. However, the same report also identifies the ease of programming as the major barrier for the adoption of WSAN technology. The effort needed to develop applications using this technology is enormous since we are talking about developing distributed applications (hundreds of sensors), taking into account the limited resources of the devices which form part of the network, i.e., the memory, computational power, energy, bandwidth, processing capabilities, limited transmission power, etc. Moreover, depending on how the devices interact with each other (how they are programmed) these resources will vary considerably [2]. All of this is much more complex if, in addition, we want to add real-time constraints [3] and achieve QoS in general [4].

Programming this kind of system has traditionally been an error-prone task since it requires programming

* Corresponding author. Tel.: +34 952 13 2865; fax: +34 952 13 1397.
E-mail addresses: ecc@lcc.uma.es (E. Cañete), hfc@lcc.uma.es (J. Chen), mdr@lcc.uma.es (M. Díaz), luisll@lcc.uma.es (L. Llopis), tolo@lcc.uma.es (B. Rubio).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2010.08.022

individual nodes, using low-level programming issues and interfacing with the hardware and the network. Therefore, the complexity of designing and implementing this kind of application makes the supply of higher-level abstractions of low-level functionality necessary in order to ease the task of the application programmer.

In the last few years, different high-level programming abstractions and middleware platforms have appeared as promising solutions to address the challenges of this kind of system. Some of these proposals are virtual machined-based ones such as Mate [5] or DAVIM [6]. TinyDB [7] and SINA [8] take a different approach and are database-based. Also we can find agent-based programming approaches like Agilla [9]. Other interesting approaches are those based on tuple channels such as TC-WSANs [10] or those based on tuple spaces like TeenyLime [11].

Recently, the Service Oriented Architecture (SOA) has been considered a good candidate for developing open, efficient, inter-operable and scalable WSAN applications [12]. In Service Oriented WSANs (SO-WSANs), node sensing and actuation capabilities are presented in the form of in-network services. Application development is simplified by providing standards for data representation, a service interface description, and for facilitating service discovery. By wrapping application functionality in a set of modular services, a programmer can specify the execution flow by simply connecting the appropriate services together. Some approaches in this vein are TinySOA [12], OASis [13], Tiny Web Services [14], Open Sensor-Rich Environments [15] and TinyWS [16].

In this paper, we propose Ubiquitous SErvices for Mote Environments (USEME [17]), a service-oriented and component-based framework to develop real-time and efficient WSAN applications using a combination of macro and node-centric programming. In this way, the developers are not only able to specify the global behavior of the application (in a platform-independent way by means of the USEME abstract language) but they also have the possibility of developing the specific services provided by the different nodes using the language in which the devices are programmed. A preliminary proposal of USEME has been presented in [18,19].

That work has been extended and improved in this paper, where the USEME framework incorporates new functionalities such as reliability, QoS constraints, a configuration template, in addition to leader management and change protocols and new visual tool features. Furthermore, a case study has been implemented to test the framework and a performance study has been carried out with the aim of analyzing the obtained results. This work is a contribution to the WSAN field in the sense that it effectively provides the following innovations:

- Combination of macro-programming and node-centric programming.
- Platform independent framework.
- High level abstract language for service definition and composition.
- Group hierarchy architecture with the possibility of defining group constraints.
- Real-time support that allows programmers to define QoS constraints in the communication between services.
- Visual tool that assists programmers to specify the global behavior of the network.

On the one hand, we propose a framework supporting a high-level service-oriented programming model, so that, the application programmers can concentrate on the application logic instead of concerning themselves with low-level issues such as messaging and routing protocols, data caches and neighbor lists. In addition, our approach takes into account QoS as a main goal, providing the application programmer with the possibility of specifying real-time constraints, reliability issues and group formation restrictions. On the other hand, the framework middleware is designed to achieve efficient implementation of the communication and coordination among nodes. Moreover, group concept is an important part of achieving scalability and energy saving.

The USEME framework is based on the concept of services, ports and operations. A service is comprised of a set of ports and each port defines the operations that can be executed in the service and whether these operations are provided or required. The USEME application is composed of different groups of nodes which isolate certain functionality, that is, only nodes in the same group can communicate with each other. Thus, services must be published within a group so that the nodes in this group can use them.

Two prototypes have been implemented in order to assess the feasibility of our approach, using Imote2.Net technology from Crossbow [20] and Sun SPOT devices [21].

The rest of the paper is structured as follows. In Section 2 the motivation of our work is presented and a case study is then described, which will be used throughout the paper together with the USEME framework as an illustrative example. Section 3 presents the USEME framework and the different parts it is composed of are explained in detail. In Section 4 the environment set-up used to test the USEME framework is described and a performance evaluation is presented. Section 5 describes an analysis of related work. Finally, some conclusions are sketched and future work is discussed in Section 6.

2. Motivation and case study

With the passing of time, the range of areas where the WSAN applications can be applied is constantly growing. We are in the presence of a relatively new, promising and powerful technology which offers a lot of advantages to develop many kinds of applications that were previously impossible. Not all are advantages however, as many parameters must be controlled. On the one hand, all WSAN applications have an important and common requirement: communication. Developers have to contend with a widely distributed application, therefore, coordination mechanisms for both sensor–actor and actor–actor interactions are needed. This communication requirement must be carefully considered by the developers before implementing WSAN applications, as both the network lifetime and scalability depend so much on it.