Development of an IoT Environmental Monitoring Application with a Novel Middleware for Resource Constrained Devices.

Article

Accepted version

S. Gaglio, G. Lo Re, G. Martorella, D. Peri, S.D. Vassallo

In Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine (MobileMed 2014)

NDS Research Group - Networking and Distributed Systems
University of Palermo, ITALY
http://www.dicgim.unipa.it/networks

Development of an IoT Environmental Monitoring Application with a Novel Middleware for Resource Constrained Devices

Salvatore Gaglio
salvatore.gaglio@unipa.it

Giuseppe Lo Re
giuseppe.lore@unipa.it

Gloria Martorella
gloria.martorella@unipa.it

Daniele Peri
daniele.peri@unipa.it

Salvatore Davide Vassallo
salvatore_davide.vassallo@unipa.it

Abstract

In this paper the development of a Mobile Health monitoring system is described. The system combines user location data with air quality information provided by a heterogeneous sensing infrastructure providing users with advises about their daily exposure to air pollutants. The highly dynamic integration of different kind of nodes, mostly characterized by rather constrained resources, of this application is crucial to implement the Internet of Things vision, and requires powerful and effective programming methodologies to abstract implementation of high-level distributed processing from hardware dependencies. We then describe our programming methodology and our novel middleware supporting distributed applications on constrained devices. Our development approach is based on distributed symbolic processing through executable code exchange among nodes, and permits to extend the capabilities of nodes even after their deployment.

1 Introduction

With the recent technological advances in wireless technologies and the growing popularity of computing devices, information processing has become ubiquitous and part of everyday activities. Common objects, such as cars, electrical appliances, toys, and clothes can be fitted with microprocessors, sensors and communication devices that allow them to be interconnected and interact with each other, offering a whole range of fascinating possibilities culminating in the vision of the "Internet of Things" (IoT) [1]. The pervasive nature of interconnected and collaborating devices of the Internet of Things has a direct application also to the medical field. Popular devices like smartphones and sensing systems can be connected to create an ubiquitous infrastructure that provides accessibility to health care services and information in order to improve individuals' quality of life. This new way to provide health services is also referred to as "Mobile-Health" [2]. Mobile-Health can be thus considered the result of the convergence of wireless communication systems, wireless sensor networks and ubiquitous computing technologies to provide healthcare services.

Following our previous research, in which we designed an application [3] able to give individuals indications about their daily exposure to air pollution, we present a much wider project that requires integration and interoperability of different networks such as mobile sensor networks and WSNs. In this paper we describe the implementation of an IoT Mobile-Health application using a novel middleware that we designed to permit high-level symbolic programming even on resource constrained devices. The remainder of the paper is organized as follows. In section 2, we give an overview about air pollution monitoring in the context of Mobile-Health. In section 3, we discuss the architecture of our urban air quality monitoring system. Our development methodology is described in section 4, along with our novel middleware and its use in the implementation of the air quality monitoring system. Then, section 5 reports our conclusions.

Figure 1: System architecture for air quality monitoring. The system is composed of heterogeneous sensing devices that measure several pollutants in the air, some access points, and a monitoring server that answers to queries about the air quality data from mobile clients.

2 Air Quality Monitoring

Air pollution in large urban areas may have a significant impact on human health and on the environment. Urban air quality is usually monitored by highly reliable networks of fixed stations. A fixed monitoring station can accurately measure a wide range of pollutants. However, permanent monitoring stations are frequently placed so as to measure ambient background concentrations or at potential hotspot locations, and they are usually several kilometers apart. Urban pollution varies spatially, as it is reasonable to expect, accordingly to human activities, topography, and local micrometeorology. The large cost of acquisition and maintenance of governative air quality monitoring stations limits the number of such facilities, resulting in a non-scalability of the system and in an extremely limited spatial resolution of the pollution maps. To overcome these problems it is necessary to adopt more pervasive monitoring systems. A considerable number of projects are described in literature that use low cost sensing devices carried by people or by mobile vehicles [4] for sensing air pollutants. In [5] authors show an environmental sensing approach that empowers citizens to reinvigorate their awareness of, and concern for, pollution. ExposureSense [6] is a rich mobile participatory sensing infrastructure able to monitor people's daily activities as well as to compute a reasonable estimation of pollution exposure in their daily life. In [7] authors demonstrate the usability of the CalFit smartphone technology to track person-level time, geographic location, and physical activity patterns for improved air pollution exposure assessment. In [8] authors introduce a cloud-based knowledge discovery system that infers real-time, fine-grained air quality information throughout a city on the basis of the air quality data reported by existing monitor stations and a variety of data collected in the urban area, such as meteorology, traffic flow, human mobility, structure of road networks, and points of interests. The system offers a mobile client, with which a user can monitor the air quality of multiple locations in a city. In [3] we presented an Android application providing users with air quality data. Exploiting smart-phones capabilities, our application implements an ubiquitous and unobtrusive monitoring system able to inform users about their daily air pollution exposure by combining user location data and urban air quality information provided by the network of fixed monitoring stations.

3 Designing an Urban Air Quality Monitoring System

In order to deploy a scalable and high-density air quality monitoring system with fine spatial and temporal scales, we propose an efficient and scalable platform that enables users to monitor their daily exposure to air pollutants by combining user location data and urban air quality information provided by a heterogeneous sensing infrastructure. As shown in Figure 1, our system is composed by mobile and fixed sensor nodes integrating the existing network of fixed stations. The sensor readings are transmitted to a back-end server that stores gathered data, ensuring their integrity, security and availability, and provides those data for several services. Exploiting low cost sensors pervasiveness and vehicles mobility, the system covers all the city area and collects detailed urban air quality data, offering a wide spatial coverage, and a fine granularity of the detected characteristics. The sensors periodically monitor the air quality and transmit, through some gateways, the collected data to a central storage system in which data are processed and made available to queries from mobile devices. Users can use their mobile clients to monitor the air quality of multiple locations in the area, for example to decide where and when to go jogging. The collected data can also help point out specific causes of air pollution. The system models the daily individual exposure to air pollutant by tagging and monitoring users' movements through smartphone GPS data [3]. By combining context-aware information, given by users' smartphones, and urban air pollution levels provided by the sensing layer, the system can give an estimation of the amount of air pollutants inhaled during the day. Differently from mainstream approaches, in which the sensor nodes are programmed conventionally, the proposed Mobile-Health platform is developed using a symbolic programming methodology and a novel middleware that we designed to program homogeneously heterogeneous nodes. Although this approach fosters high-level representation of the application tasks, the computation is effective even on resource constrained sensing devices. Our platform makes the nodes able to directly exchange executable code and supports interactive programming with iterative refinement of code even on a deployed network [9]. The network may thus acquire new capabilities at runtime. This versatility has been already proven in AmI applications [10, 11] and can be effective in other highly dynamic IoT scenarios, including Mobile-Health.

4 The Proposed Approach

The paradigm adopted to program the devices composing the sensing infrastructure of a Mobile-Health application, is often a disregarded aspect when compared to data acquisition and data analysis. In fact, the design of large-scale distributed applications, such as the air quality monitoring described in section 3, requires to interconnect heterogeneous cooperating devices. This compels the application designers to cope with the essential characteristics of networked objects that differ in architecture, capabilities, protocols, data representation, and so on. Our middleware platform [12, 13] addresses this troublesome task by providing high level abstraction tools to reduce the burden of designing and implementing applications that run on arbitrary devices. Our platform permits to:

- program the sensing infrastructure layer of a Mobile-Health application through a high-level symbolic approach;
- expand the platform itself in order to support a wide range of services and applications;
- re-program the nodes even after their deployment.

Our middleware is built atop a Forth interactive environment, permitting the designer to develop applications by expanding the language itself with high-level *words*. The programs executed by each of the node of the sensing infrastructure are thus sequence of words that can be chosen to map concisely high-level concept, somewhat approximating natural language descriptions of the tasks to be performed. This programming approach based on symbolic computation encourages the design of application-specific languages, and permits to program heterogeneous objects uniformly. Treating knowledge elements, both data or code, at a symbolic level, grants interoperability among collaborative objects since the same high-level code can be implemented on different devices, with minimal hardware dependency. As an example, nodes and stations, which are sensing devices with different architectures and resource availability, acquire a sample of carbon monoxide, by executing the same command:

```
CO sample
```

As a result, already deployed sensor nodes, as well as fixed monitoring stations, expose their knowledge to each other with the same symbolic representation. The interactions occurring in complex distributed applications take place in terms of symbols exchanged between the entities of the infrastructure layer –i.e. stations and nodes. Having an on-board interpreter, fixed stations and already deployed nodes can thus exchange symbolic programs that are executed once received as radio input.

A Forth environment on already deployed nodes makes the distributed middleware able to be incrementally extended by defining new words on the basis of already defined words. New implementations of protocols and communication requirements needed by large distributed applications as well as security protocols can be easily added. Furthermore, such peculiarity can be exploited equally by heterogeneous objects. A monitoring station can force a certain behavior on already deployed nodes, e.g. to tell a node to send an alarm when the level of a certain pollutant overcomes a specified threshold (Figure 2) as well as to provide further abilities to a node. A simple rule implementing this behavior would be:

```
CO sample threshold > [if] alarm [then]
```

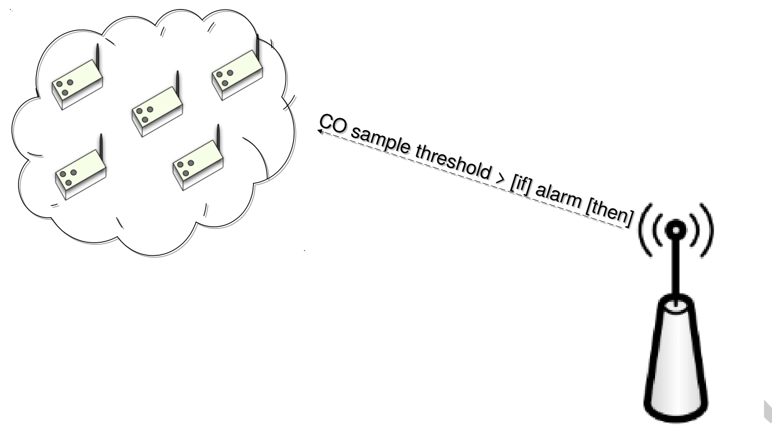

Figure 2: Our middleware allows nodes and stations to exchange executable code. Through a Forth environment executed by the nodes, it is possible to interpret messages incoming from the radio. Therefore, reactive behaviors can be easily implemented. As an example, a station tells the network to acquire a sample of carbone monoxide and to send an alarm if the CO sample exceeds a predefined threshold.

The `alarm` word defines the actions to be taken in case the event occurs. Other formalism can be added whenever the application requires them, even at runtime. We have already shown how to extend our middleware with Fuzzy Logic expressions and rules [10]. Moreover, the deployment of new devices or their replacement can be done without programming the nodes in advance, while updating the software application can be done on already deployed nodes. On the other hand, sensor nodes can be made to perform more advanced collaborative computations than sheer sensing of pollutant agents. Through the exchange of code, sensor nodes can even perform on-board air quality data fusion and aggregation tasks with high-level symbolic implementations. Symbolic reasoning can be carried out to infer collateral information in a distributed fashion, such as the identification of nodes' relative position or their membership to a particular deployment area [10], when needed. Although we are dealing with high-level behaviors, the efficiency of the implementation is not shattered because meaningful words can run directly on the target hardware. Time-critical code may also be written using the on-line assembler. Our middleware has been experimented on the Iris Mote platform that required us to implement hardware-dependent words to deal with specific features of the platform such as the radio transceiver, the on-board sensors, and the expansion boards. However, to make it run on other embedded devices, just a thin abstraction layer is required to comply with the target hardware platform. Due to the symbolic approach, hardware-abstracted words are not affected by the target platform above which they run. According to their finality, words are grouped into *wordsets* [9]. Air quality monitoring applications can thus benefit from this approach in terms of code reusability and maintenance and from the versatility it offers. Further services as result of market investigations and of user reviews can be offered to smartphone users just by expanding the software infrastructure capability at runtime. Queries by user smartphone trigger the exchange of code between the deployed nodes and between nodes and stations through the employment of high-level primitives indistinctly used by these different entities. The high-level symbolic programming and the real-time execution of incoming messages thus permit to implement and extend the middleware efficiently. This means more efficient routing protocols, reliable communication, security, any high-level data analysis and processing methods directly available to the air quality monitoring application.

5 Conclusions

We described the architecture and the development of a Mobile-Health system that enables users to monitor their daily exposure to air pollutant by combining their location data with urban air quality data coming from several sources. We showed how our development methodology based on high-level programming and symbolic processing can address some important issues arising when heterogeneous devices, often characterized by constrained resources, are requested to cooperate, which is a crucial feature of IoT applications. We then presented our novel middleware platform that supports our methodology and enables the implementation of high-level data analysis and data processing efficiently on resource constrained devices.

References

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: a Survey. *Computer networks*, 54(15):2787–2805, 2010.

[2] Upkar Varshney. *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. Springer, 2009.

[3] G Lo Re, D. Peri, and S. D. Vassallo. A Mobile Application for Assessment of Air Pollution Exposure. In *Mobile and Information Technologies in Medicine and Health 2013*, 2013.

[4] Giuseppe Lo Re, Daniele Peri, and Salvatore Davide Vassallo. Urban Air Quality Monitoring Using Vehicular Sensor Networks. In *Advances onto the Internet of Things*, pages 311–323. Springer, 2014.

[5] Radka Peterová and Jakub Hybler. Do-it-yourself Environmental Sensing. *Procedia Computer Science*, 7:303–304, 2011.

[6] Bratislav Predic, Zhixian Yan, Julien Eberle, Dragan Stojanovic, and Karl Aberer. Exposuresense: Integrating daily activities with air quality using mobile participatory sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 303–305. IEEE, 2013.

[7] Audrey de Nazelle, Edmund Seto, David Donaire-Gonzalez, Michelle Mendez, Jaume Matamala, Mark J Nieuwenhuijsen, and Michael Jerrett. Improving Estimates of Air Pollution Exposure through Ubiquitous Sensing Technologies. *Environmental Pollution*, 176:92–99, 2013.

[8] Yu Zheng, Xuxu Chen, Qiwei Jin, Yubiao Chen, Xiangyun Qu, Xin Liu, Eric Chang, Wei-Ying Ma, Yong Rui, and Weiwei Sun. A Cloud-Based Knowledge Discovery System for Monitoring Fine-Grained Air Quality. Technical report, MSR-TR-2014-40, 2014.

[9] Salvatore Gaglio, Giuseppe Lo Re, Gloria Martorella, and Daniele Peri. A Fast and Interactive Approach to Application Development on Wireless Sensor and Actuator Networks. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pages 1–8, Sept 2014.

[10] S Gaglio, G Lo Re, G Martorella, and D Peri. High-level Programming and Symbolic Reasoning on IoT Resource Constrained Devices. In *Accepted at The First International Conference on Cognitive Internet of Things Technologies (COIOTE 2014)*, 2014.

[11] S Gaglio, G Lo Re, G Martorella, and D Peri. Programming Distributed Applications with Symbolic Reasoning on WSNs. In *Accepted at International Conference on Computing, Networking and Communications (CNC2015)*, 2014.

[12] Salvatore Gaglio, Giuseppe Lo Re, Gloria Martorella, and Daniele Peri. A Lightweight Middleware Platform for Distributed Computing on Wireless Sensor Networks . *Procedia Computer Science*, 32(0):908 – 913, 2014. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014).

[13] Distributed Computing for Resource Constrained Devices (DC4CD). `http://www.dicgim.unipa.it/networks/ndslab/rl_dc4cd.php`, 2014.