

Chapter 7

Final Remarks

Abstract In this chapter we summarize the contributions of this Book to advance the state-of-the-art in the development of IoT applications. The main contributions are: (i) to present a middleware-layer infrastructure, SmartSensor, that focuses on the integration of Wireless Sensor Networks (WSN) in the Web, providing WSN as a service, accessible as any other Web resource and enabling the discovery and composition of services to form Web mashup applications; (ii) to describe step by step the development of an application using the SmartSensor infrastructure, the Parking Lot Application, that helps drivers to find available parking space in a building. SmartSensor is composed of three software modules: (i) the Sensor Integration Module (SIM) integrates the distinct WSN devices and provides a RESTFUL interface to access them as Web resources; (ii) the Programming and Execution Module (PEM) enables the composition of value-added services from multiple Web sources; (iii) the Web 3.0 Integration Module (WIM) integrates the resources with Web 3.0 applications and platforms.

Keywords Internet of Things (IoT) · Web of Things (WoT) · Wireless sensor networks (WSN) · Middleware for IoT · SmartSensor · Parking lot application

7.1 Introduction

The Internet of Things (IoT) represents a new direction on the use of computers in everyday life as it enables the integration of the Internet with the physical world, populated with sensors, actuators, and embedded communication hardware. The heterogeneous physical devices become a part of the Web and can be accessible using the well-known Internet protocols. This scenario arises the opportunity for creating sophisticated applications based on physical world data, such as monitoring inaccessible and remote spaces (oil platforms, forests, tunnels, mines, pipes), environmental

monitoring (earthquakes, floods, radiation areas, fire, rainfall) [6], smart cities, smart buildings, smart homes, health-care, to name a few.

Some middleware platforms already exist, addressing one of the main challenges to enable realization of IoT: interoperability across heterogeneous resources to relieve the application developer from the burden of integrating them. However, according to [5], the existing proposals do not cover the full set of functionalities to meet the requirements of an IoT middleware. In addition, most of them are under development and did not reach the stage of maturity to be adopted in a large scale.

This Book described a middleware-layer infrastructure, SmartSensor, that focuses on the integration of Wireless Sensor Networks (WSN) in the Web, providing WSN as a service, accessible as any other Web resource and enabling the discovery and composition of services to form Web mashup applications. In fact, this infrastructure targets a significant barrier to the widespread use of WSNs, that is the complexity of applications development that need to deal with low-level concerns of WSN. One of the fundamental characteristic of SmartSensor is to be based on the REST (Representational State Transfer) principles [2] and to use Web standards and protocols, such as URIs (Uniform Resource Identifier) and HTTP (Hypertext Transfer Protocol). This is especially useful for the integration of WSN to the Web, since the heterogeneous nature of their devices. In SmartSensor, WSNs are connected to the Web through a gateway node and their data are exposed as RESTful Web resources. Therefore, the interactions occurs in a RESTFull style by using the set of simple, well-defined HTTP main operations (GET, POST, PUT and DELETE i.e. the verbs of REST). Currently, SmartSensor supports three different platforms that are proper to a WSN scenario typically endowed with several tiny and low-power networked devices in interaction.: Arduino,¹ Sun SPOT,² and motes from the MICA family (produced by MEMSIC, former Crossbow),³ based on the TinyOS operating system,⁴ specially designed to sensors.

The infrastructure provided by SmartSensor to seamless integrate the resources relies on three main building blocks:

1. The Sensor Integration Module (SIM) integrates the distinct WSN devices and provides a RESTfull interface to access them as Web resources. It enables data collection and delivery and supports publish and discovery of WSN services. SIM components are deployed both in sensor nodes and in gateways nodes.
2. The Programming and Execution Module (PEM) enables the composition of value-added services from multiple Web sources, mixing public available services and those provided by SmartSensor that are registered in SIMs. PEM offers a DSL (Domain Specific Language) based on the Enterprise Mashup Markup Language (EMML) [1] and an interpreter for such DSL. It provides mechanisms to create, interpret, and execute Web mashups specified via EMML scripts.

¹ <http://www.arduino.cc/>

² <http://www.sunspotworld.com/>

³ <http://www.memsic.com/>

⁴ <http://www.tinyos.net/>

3. The Web 3.0 Integration Module (WIM) integrates the resources with Web 3.0 applications and platforms.

This Book also detailed the development of a parking lot application that consists of a WSN-based vehicle detection system connected to the SmartSensor infrastructure. The three distinct phases involved in the development of such an application were discussed: (i) the hardware configuration of the sensor nodes and the sink node; (ii) the programming of the components to collect and interpret the signals collected by the WSN ultrasonic sensors; (iii) the programming and installation of the PEM mashup application to monitor the parking lots registered in the infrastructure. This application was used in three parking lots located at the Federal University of Rio de Janeiro (UFRJ).

7.2 Contributions

The main contributions of this work are:

- to present an overview on the emergent WoT paradigm, describing its main concepts, principles and building blocks;
- to describe a middleware-layer infrastructure that integrates distinct WSN endowed with a myriad of heterogeneous elements (sensors, actuators, communication hardware);
- to provide a RESTful-based programming model that hides, from the developer, the low-level details to integrate different types of WSN sensed data in a Mashup application. Using this programming model, developers can create applications without having specific knowledge about physical devices or networking environments;
- to describe, step by step, how to create a Web Mashup application using the SmartSensor infrastructure integrated with a widely used EMMML graphical editor and runtime environment, Presto.⁵ It is worthwhile to mention that SmartSensor can be used with any other tool based on EMMML.

7.3 Future Work

Despite the rising popularity of IoT and the development of middleware solutions for facilitating the development of mashup applications, some important aspects need to be further addressed for making IoT a reality. The main functional components of an IoT middleware, stated by [5], includes interoperation, context detection, device discovery and management, security and privacy, and managing data volume. Although SmartSensor can be considered as a middleware for IoT, in its current version, it only

⁵ <http://www.jackbe.com/prestodocs/v3.2/presto-intro/prestoIntro.html>

provides communication and integration services as well as a programming model to develop applications on top of the WSN infrastructure.

Managing high data volume and supporting security and privacy are the two key capabilities to be considered in next versions of SmartSensor. According to [6], IoT generates a huge amount of data, yielded by physical and virtual elements connected to the Internet, and the data volume needs to be stored and easily reached. The convergence of Cloud Computing with IoT has been pointed out as a promising approach [3] to address this issue by establish a cloud-based IoT environment, the so-called *cloud-of-things* [4].

Another important issue that deserves a further investigation, highlighted by [7], is the support for dynamic adaptation of mashups according to the user requirements and the runtime environment. This aspect needs a combination of context monitoring and management, adaptive behavior modeling, and the realization of dynamic adaptation of the mashup.

References

1. Enterprise Mashup Markup Language, Open Mashup Alliance, 2010. Available at <http://www.openmashup.org/>
2. Fielding, R. (2000). *Architectural styles and the design of network-based software architectures*. Doctoral dissertation, University of California, Irvine.
3. Gao, L., Zhang, C., & Sun, L. (2011). *RESTful web of things API in sharing sensor data*. In the 2011 International Conference on Internet Technology and Applications (iTAP), IEEE.
4. Soldatos, J., Serrano, M., & Hauswirth, M. (2012). Convergence of utility computing with the internet-of-things. *proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous, Computing (IMIS'12)* (pp. 874–879).
5. Bandyopadhyay, S., Sengupta, M., Maiti, S., & Dutta, S. (2011). Role of middleware for internet of things: A study. *International Journal of Computer Science and Engineering Survey*, 2(3), 94–105.
6. Vermesan, O., Friess, P., Guillermin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I., Mazura, M., Harrison, M., Eisenhauer, M., Doody, P. (2011). *Internet of Things Strategic Research Roadmap*. Global Technological and Societal Trends. (pp. 9–52). Aalborg: River Publishers.
7. Zeng, D., Guo, S., & Cheng, Z. (2011). The web of things: A survey. *Journal of Communications*, 6(6), 424–438. Sep 2011 doi:10.4304/jcm.6.6.424-438.