Application Development for the Internet of Things: Observations and Challenges

Ajay Chhatwal, Kaushal Shukla, Animesh Pathak

Application Development for the Internet of Things: Observations and Challenges

Ajay Chhatwal, K. K. Shukla
Department of Computer Engineering
IIT (BHU), Varanasi
Varanasi, India
chhatwal.ajay.1990@gmail.com,
kkshukla.cse@itbhu.ac.in

Animesh Pathak
Inria Paris-Rocquencourt
France
animesh.pathak@inria.fr

***Abstract*— The Internet of Things (IoT) domain can be considered to be an amalgamation of the hitherto well-researched Wireless Sensor and Actuator Networks (WSAN) and Pervasive Computing domains. However, application development on this platform still remains challenging. In this paper, we first discuss a specific real world instance of an oft-cited IoT application: the Smart Home application, and gain insight into IoT application development by actually implementing it.**

Based on the above, we then present the design, development and deployment techniques for a real world IoT system from ground up and describe of various interaction patterns that naturally occur in such applications. We further discuss the challenges faced while reusing and combining approaches from the existing domains of WSANs and Pervasive Computing, into the domain of IoT. Finally, we conclude by using these insights to present a roadmap for designing an application development framework for IoT.

Keywords- Internet of Things, Pervasive Computing, Wireless Sensor Networks

I. INTRODUCTION

The Internet of Things (IoT) domain has been discussed in literature for some time now with several similar but non-identical definitions. We use here the definition given by the CASAGRAS project [1]: “A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability.”

This definition, as well as others that have been used in other funded research reports, such as CERP (Cluster of European Research Project) - IoT [2], emphasizes the internetworking between heterogeneous “smart” devices such as sensors, actuators, computers and smart phones etc., and the use of services over the internet. Any application development framework for the IoT, therefore, needs to support all of these heterogeneous devices.

It is interesting to note that though the field of IoT in itself is relatively new and evolving, the component systems themselves are sufficiently mature. There has been a large amount of research in the related fields of Wireless Sensor and Actuator Networks (WSANs) and Pervasive Computing, which are now well established fields with large number of groups working on them, in both academia and industry.

However, current work in the field generally concentrates either on application development for WSANs [3], [4] or Pervasive Systems [5], [6] alone, but not both simultaneously. The few projects that do explore the integration of WSANs with pervasive systems, envisage WSAN as a discrete component of the system, accessed via its base station as a Web Service [7], or other mechanisms such as the base station (agent) tuple space [8]. A study of application development techniques for IoT is a necessary first step for designing effective application development frameworks for it.

In this paper, we explore how the Internet of Things domain can be thought of as the composition of the existing domains of WSANs and Pervasive Computing. The main contributions of this paper are the design, development and deployment techniques for a real world IoT system from ground up, description of various interaction patterns that naturally occur in such Internet of Things applications, and an insight into the challenges faced while reusing and combining approaches from the existing domains of Wireless Sensor and Actuator Networks (WSANs) and Pervasive Computing, into the domain of IoT.

II. SMART HOME- TEMPERATURE MANAGEMENT SYSTEM

In literature, a class of IoT applications that comes up frequently is the smart home [9], featuring prominently in surveys such as [10]. It primarily features a system that controls the devices in the house, such as lights, thermostats etc., either automatically or by user intervention through applications running on smart phones [11] or through RFID readers [12].

Figure 1 describes the design of the actual prototype system developed by us - A *Temperature Management System* controlled by smart phone or RFID smart card. In each room, a *temperature sensor* samples the temperature periodically. The sampled readings are aggregated and processed at a *Temperature Processing Server*, which also doubles as a server for requests from a *RFID Card Reader* or a *Smart Phone*. The Temperature Processing Server processes the aggregate sensor reading and user commands and produces an actuation action, which is then sent to the *Dummy Actuator*. The room temperature preferences of different residents of the home are stored with the *Preferences Store Service*. The Temperature Sensor samples temperature every 10 seconds, while the Dummy Actuator causes the LED to glow in different colors based on the commands HEAT, COOL, or STOP.

For our implementation and deployment, we have used the following hardware: SUN SPOT sensors/actuators/base stations, Android based smart phones, JavaCard API compliant RFID cards & card readers and Laptops equipped with the Java Runtime Environment. IEEE 802.15.4, TCP/IP and SOAP/HTTP protocols are used at various layers for communication between these heterogeneous devices. The translations between the messages sent using different protocols are performed at the application layer in each of the applications running on these devices.

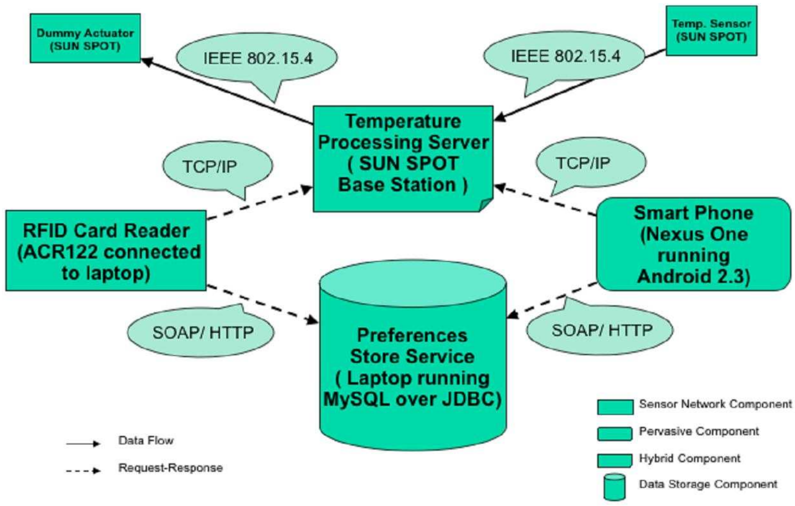

Figure 1. Schematic diagram of Temperature Management System Prototype

The sensing and actuating portions of the system have been developed as SUN SPOT "free-spot" (wireless sensor/actuator) applications, the Temperature Processing Server as a SUN SPOT "host-spot" (base station) application, the GUI based temperature query/control as an Android application, the RFID Card Reader based temperature control as a Java desktop application, and the Preferences Store as a SOAP based web service. We have released the code for each of these different portions of the application under open source license at http://code.google.com/p/temperature-management-system/.

The four use cases of the system are:

1. *Automatic Temperature Management*: When no user is interacting with the system, it automatically maintains the temperature at a pre-defined default value.

2. *Temperature Query via Smart Phone*: A user can query the temperature of a room by using a smart phone application.

3. *Temperature Control via Smart Card*: A user swipes a smart card to change the temperature of a room to his/her preference.

4. *Temperature Control via Smart Phone*: A user changes the temperature of a room by using a smart phone application.

III. Interaction Patterns in Internet of Things Applications

A. *Hierarchical Data Aggregation*

An IoT system often features sensor nodes, which sample a physical quantity, and typically needs to perform aggregation of the sensor readings in order to get reliable estimates of the physical quantity being sensed. Most commonly, this is done by using a hierarchy of data aggregation nodes. The Temperature Management System employs a single level of hierarchy based on location of the Temperature Sensors, with all the temperature sensor readings within radio range being collected at the Temperature Processing Server present in the room and the aggregate value representing the temperature of the room as a whole.

This kind of behavior is prevalent in traditional WSAN systems [13], where a multilevel hierarchy is naturally evident from the (static) architecture of the system.

B. *Actuation Driven by Sensors*

An IoT system often features actuators being driven by one or more sensors. A change in the sensor reading can bring about the change in the action performed by the actuators. In the Temperature Management System, a change in the reading from any of the Temperature Sensors can change the aggregate room temperature and move it across the desired temperature (threshold) set by the user, driving the Actuator to heat or cool the room accordingly.

This kind of behavior is prevalent in traditional WSAN systems [13], and forms the *sense-compute-actuate* loop.

C. Request-Response

An IoT system often features queries, originating from some part of the system, which trigger a response either from another part of the system or from an external system. In the Temperature Management System, the smartphone queries the Temperature Processing Server for the aggregate room temperature, and both the smartphone and the RFID Card reader query the Preferences Store Web Service for temperature preference of the user.

This type of behavior is prevalent in pervasive systems [14], where devices often query data from various sources within and outside the system.

D. Actuation Driven by User

An IoT system often exhibits user driven actuation, in which a user action can bring about the change in the action performed by the actuators. In the Temperature Management System, the user action such as changing the desired temperature preference via smartphone can drive the Actuator to heat or cool the room accordingly.

The Internet of things domain makes it possible for *users to influence and configure the actions taken by the actuators*, and thus directly influence the environment. This is an emergent property of the combination of WSAN and Pervasive subsystems.

IV. CHALLENGES

Upon comparison of our reference system □Temperature Management System, with traditional WSAN and Pervasive systems respectively, several differences are observed, challenging the integration of these two sub domains of IoT into a single entity.

A. Comparison with Traditional WSANs

1) Request- Response Interaction Pattern

Traditional WSANs generally rely on one-way communication i.e. a sensor produces some data and sends it to a central aggregator while an actuator performs the desired function when it receives a command from a central decision making node. However, a pervasive system needs to support the request response interaction pattern.

This pattern is observed in the Temperature Management System when the Smartphone and RFID Card Reader interact with the Preference Store Service to get user temperature preference. The pattern is also observed when the Smartphone and RFID Card Reader interact with the Temperature Processing Server.

2) GUI on Device

Traditional WSANs are primarily data processing systems. They aggregate data and then based on the aggregate value, perform some action (actuation). GUIs, if present at all, are used only for configuring the system. In contrast, the typical device in a pervasive system is a Smartphone or a PDA, which essentially has a GUI. In fact, the user interacts with the pervasive system using the GUI, which is essential to the very nature of pervasive computing.

The Smartphone GUI in Temperature Management System allows the user to view and control the temperature of the room.

3) Mobility of Nodes

Traditional WSANs typically have fixed static configurations. The nodes are logically, if not physically, linked to one or more physical region, such as room or a floor. This is generally known as the logical scope of the node. However, pervasive systems are characterized by the mobility of the nodes, which are free to move about, as long as they are connected to the network. There is generally no affinity attached to any particular region by a pervasive node.

The Smartphone in Temperature Management System is a mobile node and is not logically attached to any region. Before being allowed to control the temperature of a particular room, the Smartphone user needs to authenticate himself/herself as an authorized resident for the room.

B. Comparison with traditional Pervasive systems

1) Periodic Generation of Large Amount of Data

In traditional pervasive systems, data is typically generated sporadically, typically on user initiation. The user interacts with the GUI on the device, which then generates and/or requests data on the device memory, or over the network. In contrast, a WSAN is characterized with the periodic generation of large amount of data, which is used for sampling the environment at multiple locations.

The Temperature Sensors in the Temperature Management System generate the temperature reading at the rate of one reading per 10 seconds.

2) Aggregation of Similar Data

Traditional pervasive systems usually exhibit only the one-to-one communication pattern. However, WSANs typically use many-to-one and one-to-many communication , with data generated by many sensors, data aggregated at a processing node and action decided on basis of some computation, and message conveying the action sent to many actuators.

The Temperature Sensors in the Temperature Management System generate the temperature reading, which is then aggregated at the Temperature Processing Server and the desired action (HEAT, COOL, or STOP) is conveyed to all actuators within the room.

3) Power Management

Both pervasive systems as well as WSANs consist of nodes which typically run on battery power. However, energy efficiency is a much bigger concern for WSAN components like sensors and actuators, compared to pervasive nodes like smart TV that are powered, or Smartphones which are re-charged by users regularly.

The temperature sensors in the Temperature Management System sleep after generating the temperature reading, thereby conserving battery power.

C. *The Problem of Network Stack Interoperability*

The heterogeneous devices comprising The Internet of Things support different networks stacks, depending on its type, cost and application. Typical networks include IEEE 802.11 (WLAN), IEEE 802.15.4, Bluetooth, GPRS, EGDE, UMTS, etc.

Any effort on uniting these heterogeneous devices into a single system needs to take into account the nature and interoperability of the networks supported by the devices. It may be necessary to reformat messages and adapt routing based on the type of sending and receiving devices.

The Temperature Management System currently circumvents this problem by localizing the techniques for interoperability between WSAN and pervasive components at the Temperature Management Server node. However, this is not practical for large systems.

V. Conclusion and Future Plans: Towards a Framework for IoT Application Development

We can conclude that the Internet of Things domain can broadly be considered as a composition of the existing domains of Wireless Sensor and Actuator Networks and Pervasive Computing. However, we observe that some newer interaction patterns emerge in IoT systems, and there are significant challenges in reusing and combining existing approaches in these two domains.

Therefore, there is a need for the development of an application development framework for such applications, perhaps adapting and reusing the existing concepts and components. We believe that development of such a framework can be done effectively in the following steps.

A. *Development of a Domain Specific Language for the Internet of Things*

In order to develop a framework for application development for the Internet of Things, we first need to model networked devices into more general classes than found in current frameworks and toolkits, which primarily focus on specialized (WSAN and pervasive) subclasses of such devices.

B. *Adapting an Existing Software Toolkit to a More General Class of Devices*

An existing toolkit for either WSANs or Pervasive systems needs to be modified, on the basis of the model obtained above, to support both Pervasive and WSAN nodes, and possibly even other classes of devices. In the event that this is not possible, a new toolkit needs to be developed.

C. *Development of Abstractions for Incorporating Existing Internet Components into Internet of Things Systems*

From the various definitions of the Internet of Things, it is clear that existing internet components, especially Web Services, have a role to play in this domain. Techniques need to be developed for interacting with such existing systems, employing techniques like client generation for SOAP web service based on WSDL.

D. *Integrating Dynamic Discovery of Devices and Services*

An IoT framework should allow for a system to evolve at runtime. Approaches that allow for dynamic detection of heterogeneous devices and their services need to be investigated and implemented in the framework.

Acknowledgment

This work has been funded in part by the French ANR MURPHY project.

References

[1] CASAGRAS, "CASAGRAS EU project final report", http://www.grifs-project.eu/data/File/CASAGRAS%20FinalReport%20(2).pdf.

[2] CERP-IoT, "CERP-IoT report - Internet of Things Strategic Research Roadmap", http://www.grifs-project.eu/data/File/CERP-IoT SRA_IoT_v11.pdf.

[3] Ryan Newton, Greg Morrisett, and Matt Welsh, "The Regiment macroprogramming system," *Proc. 6th Int'l Conf. Information Processing in Sensor Networks* (IPSN '07), ACM , 2007, pp. 489-498.

[4] Animesh Pathak and Mahanth K. Gowda, "Srijan: a graphical toolkit for sensor network macroprogramming," *Proc. 7th Joint Meeting European Software Eng. Conf. and ACM SIGSOFT Symp. Foundations of Software Engineering* (ESEC/FSE '09), ACM, 2009, pp. 301-302.

[5] Sonia Ben Mokhtar, Nikolaos Georgantas, and Valerie Issarny, "COCOA: COnversation-based service COmposition in pervAsive computing environments with QoS support.," *J. Systems Software.,*vol. 80, no. 12, Dec. 2007, pp. 1941-195.

[6] Wilfried Jouve, Julien Lancia, Nicolas Palix, Charles Consel, and Julia Lawall, "High-level programming support for robust pervasive computing applications," *Proc. 6th IEEE Conf. Pervasive Computing and Communications* (PERCOM '08). IEEE CS, 2008, pp.252-255.

[7] The OSIRIS EU Project; http://www.osiris-fp6.eu/

[8] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, Amy L. Murphy, and Gian Pietro Picco, "TinyLIME: bridging mobile and sensor networks through middleware," *Proc. 3rd IEEE Conf. Pervasive Computing and Communication* (PERCOM '05). IEEE CS, 2005, pp. 61-72.

[9] Alex S. Taylor, Richard Harper, Laurel Swan, Shahram Izadi, Abigail Sellen, and Mark Perry , "Homes that make us smart," *Personal Ubiquitous Computing*, vol. 11, no. 5, June 2007, pp. 383-393.

[10] Luigi Atzori, Antonio Iera, and Giacomo Morabito, "The Internet of Things: a survey," *Computer Networks,* vol. 54, no. 15. Oct. 2010, pp. 2787-2805.

[11] Jeffrey Nichols and Brad A. Myers, "Controlling home and office appliances with smart phones," *Pervasive Computing, IEEE* , vol.5, no.3, July 2006, pp.60-67.

[12] Mohsen Darianian and Martin Peter Michael , "Smart home mobile RFID-based Internet-of-Things systems and services," *Proc. Int'l Conf. Advanced Computer Theory and Eng.* (ICACTE '08). IEEE CS, 2008, pp. 116-120.

[13] Animesh Pathak, Luca Mottola, Amol Bakshi, Viktor K. Prasanna, and Gian Pietro Picco, "Expressing sensor network interaction patterns using data-driven macroprogramming," *Proc. 5th IEEE Int'l Conf. Pervasive Computing and Communications Workshops* (PERCOMW '07). IEEE CS, 2007, pp.255-260.

[14] Christian Becker, Gregor Schiele, Holger Gubbels, and Kurt Rothermel, "BASE - a micro-broker-based middleware for pervasive computing," *Proc. 1st IEEE Conf. Pervasive Computing and Communication* (PERCOM '03), IEEE CS, 2003 pp. 443- 451.