Smart Home: Integrating Internet of Things with Web Services and Cloud Computing

Moataz Soliman[1], Tobi Abiodun[1], Tarek Hamouda[1], Jiehan Zhou[1,2], Chung-Horng Lung[1]

[1]Department of Systems and Computer Eng.
Carleton University, Ottawa, Ontario, Canada

[2]Department of Computer Science and Eng.
University of Oulu, Finland

***Abstract*—Smart Home minimizes user's intervention in monitoring home settings and controlling home appliances. This paper presents an approach to the development of Smart Home applications by integrating Internet of Things (IoT) with Web services and Cloud computing. The approach focuses on: (1) embedding intelligence into sensors and actuators using Arduino platform; (2) networking smart things using Zigbee technology; (3) facilitating interactions with smart things using Cloud services; (4) improving data exchange efficiency using JSON data format. Moreover, we implement three use cases to demonstrate the approach's feasibility and efficiency, i.e., measuring home conditions, monitoring home appliances, and controlling home access.**

***Keywords-* Smart home, Internet of Things, Cloud Computing, Arduino, Zigbee, JSON**

I. INTRODUCTION

Smart Home (SH) promises the potentials for the user to measure home conditions (e.g., humidity, temperature, luminosity, etc.), manipulate home HVAC (heating, ventilation and air conditioning) appliances and control their status with minimum user's intervention. Researchers and practitioners have made a great deal of efforts in facilitating the concept. For example, for smart home management, Son et al. [1] proposed a resource-aware management system using a domain-object hierarchical model for representing home context. Technically, they utilized Web Services Description Language (WSDL) and Simple Object Success Protocol (SOAP) to enable remote access of home information using mobile devices. For efficient energy management, Han et al. [2] suggested a new Smart Home Energy Management System (SHEMS) based on IEEE802.15.4 and ZigBee. They designed a SHEMS-based multi-sensing and light control application for reducing total energy cost. Concerning "home nature" of a smart home in serving its users, Wu et al. [3] presented a framework to model the interaction relationship among services, spaces, and users, and fulfill the human-centric interaction requirement of a smart home. Using the framework, they developed two pervasive applications of "Media Follow Me (MFM)" and "Ubiquitous Skype." To predict the user activity, Alam et al. [4] proposed an algorithm, called sequence prediction via enhanced episode discovery (SPEED). SPEED classes the user behavior into distinct episodes, which consist of sequence activities. Based on human activity patterns, they used SPEED to extract episodes and user actions represented in a finite-order Markov model and the partial matching (PPM) algorithm for improving prediction accuracy. Chen et al. [5] also introduced a knowledge-driven approach to real-time, continuous activity recognition based on multi-sensor data streams in smart homes. The approach consists of context ontologies modeling, the situation formation process, and a knowledge-driven activity recognition architecture. In this way, activity recognition can be conducted from low-level sensor data collection, middle-level data fusion, to high level activity recognition. The SemWeb semantic technologies was used for the creation, management, and query of the semantic data. The Euler inference engine [6] was used for semantic reasoning.

The aforementioned research efforts focus on the SH features of context-awareness, energy efficiency, natural interaction, and user activity recognition. This paper proposes an approach to facilitate SH implementations. The approach integrates the emerging concepts of Internet of Things (IoT) and Cloud computing. IoT embeds computer intelligence into home devices and provides the user with a convenient way to measure home conditions and monitor home appliances. Cloud computing provides scalable computing and storage power for developing, maintaining, and running home services. In addition, using cloud computing allows the user to access (monitor and/or control) home devices anytime and anywhere.

The remainder of the paper is organized as follows. Section II presents our approach of integrating IoT and Cloud computing and selected use case design. Section III details the technical solutions. Section IV presents three use case studies respectively. And conclusion is drawn in Section V.

II. SYSTEM ARCHITECTURE AND USE CASE DESIGN

A. System Architecture for Smart Home

The system architecture for Smart Home must fulfill the requirements of measuring home conditions, processing instrumented data, and monitoring home appliances. Our approach utilizes microcontroller-enabled sensors for measuring home conditions and microcontroller-enabled actuators for monitoring home appliances in the front end. Our approach utilizes PaaS (Platform as a Service) and SaaS (Software as a Service) in Cloud computing for processing data

978-0-7695-5095-4/13 $31.00 © 2013 IEEE
DOI 10.1109/CloudCom.2013.155

at the back end. Fig. 1 illustrates the system architecture for Smart Home. It consists of the following major components:

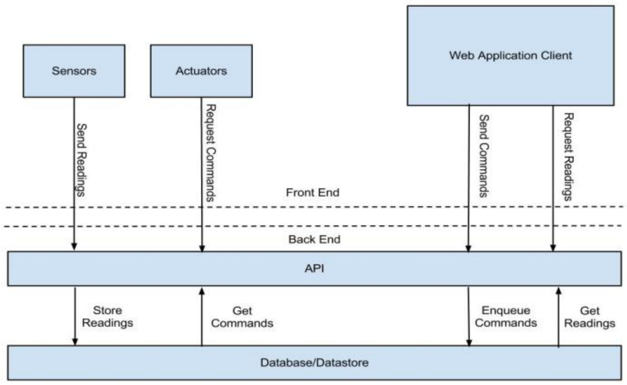

Fig. 1. System architecture for Smart Home

- Microcontroller-enabled sensors: measure home conditions; the microcontroller interprets and processes the instrumented data.
- Microcontroller-enabled actuators: receives commands transferred by the microcontroller for performing certain actions. The command are issued based on the interaction between the microcontroller and Cloud services.
- Database/Data Store: stores data from microcontroller-enabled sensors and Cloud services for data analysis and visualization, and serves as command queue being sent to actuators as well.
- Server/API layer between the back end and the front end: facilitates processing the data received from the sensors and storing the data in database. It also receives commands from the web application client to control the actuators and stores the commands in database. The actuators make requests to consume the commands in the database through the serer.
- Web application serving as Cloud services: enable to measure and visualize sensor data, and control devices using a mobile device (e.g., smart phone).

B. Use Case Design

In light of the above system architecture, we design the following three uses cases (see Fig. 2).

- Measuring home conditions. This is one of the primary use cases. It describes the action which the user performs to get the readings of home conditions, e.g., temperature and humidity.
- Managing home appliances. The use case indicates that the user is able to manipulate various household appliances, such as lights, doors, fans and air-conditioners.
- Controlling home access. This use case indicates that the user can use RFID cards to gain access to a home and have their activity logged. This would allow tracking the user's presence in a home.

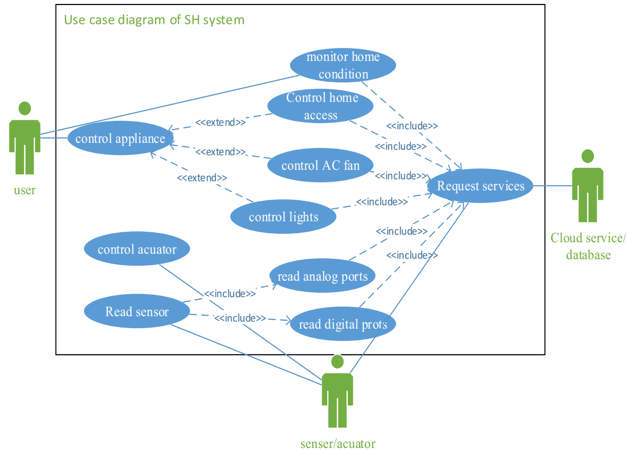

Fig.2: Use cases diagram

III. TECHNICAL SOLUTIONS

Several technologies are involved to realize our design. These technologies include Arduino microcontrollers, ZigBee communication protocol, JSON for data exchang, and Google App Engine for Cloud computing. The following subsections describe each technology in details.

A. Using Arduino for Programming Internet of Things

IoT is making computing and communication pervasive, mobile and wearable, thanks to many microcontrollers and microcontroller platforms [7] [8]. Arduino [7] consists of a programming environment and Arduino boards. The Arduino programming environment facilitates the developer to manage, compile, upload, and simulate programs. Currently, there are several Arduino boards with different performance parameters, such as the size of RAM or flash memory, the number of analog pins or digital pins as well as clock speed. Arduino platform has the main features of low cost, cross-platform, simple programming environment, open source, and extensible software and hardware. In our development, we use Arduino platform for prototyping the above designed use cases and Arduino UNO as the microcontroller board to program various types of sensors/actuators and communication technologies such as RFID and ZigBee.

B. Using ZigBee to Network Smart Home

The deployed Arduino-embedded devices (smart things) in home need to communicate each other in order to exchange sensor outputs, triggers, status messages, etc. ZigBee is one of the widely used advanced wireless technologies for establishing communication between things for home automation [9] [10]. ZigBee is a radio frequency (RF) communications standard based on IEEE 802.15.4. A Zigbee-based network usually consists of a Zigbee coordinator and Zigbee nodes. The Zigbee coordinator is responsible for creating and maintaining the network. The Zigbee coordinator manages each Zigbee node, such as washing machine, television, lamp, etc., in the network. All communications between Zigbee nodes propagate through the coordinator to the destination node. The maximum ZigBee data rate is about 250kbps and communication range can vary from 100m to 1km depending on the output power. As 40kbps can meet the requirements of most control systems, it is sufficient for controlling most home automation devices [9]. Fig. 3 presents our proposed ZigBee-networked smart home. There are two Arduino UNO microcontroller boards. The first board serves as the central receiver which is connected to all the actuators in the system and connected to the database server on the Internet through the Ethernet connection. The second board is the central transmitter that is connected to all the sensors in the system. ZigBee technology is used for communication between ZigBee sensors/actuators and the central Arduino boards.

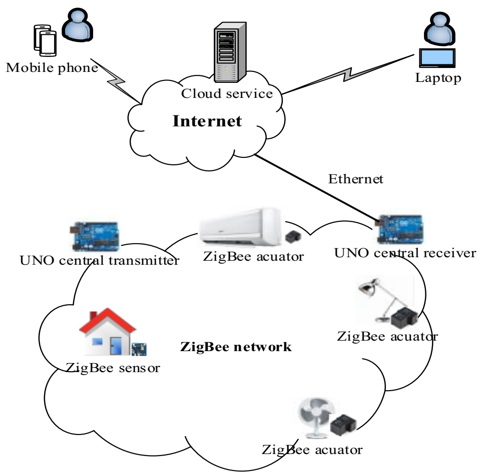

Fig. 3. ZigBee home network

C. *Using JSON for Data Exchange*

In order to communicate with the Cloud, Smart Home needs access to the Internet. Future Smart Home promises to contain a high number of sensors, many of which are constrained and have limited processing power, memory capacity and communication bandwidth. XML is widely used as message serialization format in Web-based heterogeneous systems. However, resource-constrained IoT nodes cannot always afford to perform full-fledged XML processing due to their limitations on resources [11]. JSON is a lightweight data representation syntax for storing and exchanging text information. It is much like XML, but smaller than XML, and faster and easier to parse. JSON syntax is a subset of object notation syntax. JSON data is written as name/value pairs. Building JSON into JavaScript gives it a distinct advantage over other serialization formats such as XML when working with applications partially written in JavaScript [12].

In our approach, we use several sensors for measuring home conditions. The sensors readings need transmit to the central server periodically. Meanwhile the actuator node receives commands from the central server to control home appliances. To reduce traffic load and bandwidth usage, we use JSON for data representation and exchange. The current implementation consists of 4 Arduino UNO boards. Each one is assigned an identity number from 1 to 4. A sample JSON packet format is given as the following {“source”, “4”, “destination”: “2,”token”: “1234”, “temp”: “27”, “humidity”: “120”, “proximity”: “2207”, “ambient”: “42”}, which is sent from the central transmitter board via ZigBee to the central receiver board. The central receiver board is connected to the Internet and Cloud services through the Ethernet connection. In the sample message, the first field shows the ID of the source board followed by the ID of the destination board. The token field is an authentication mechanism used to verify that the message is sent from one of the system boards. The remaining fields carry other information, e.g., temperature and humidity. JSON messages are sent from the central transmitter board to the attached ZigBee via serial connection. The ZigBee broadcasts the message which will be captured by the destination board specified in the destination field.

D. *Web Applictins with Cloud Computing*

Our architecture is based on Cloud computing which provides storage and computing resources to implement Web applications. More importantly, Cloud services can be accessed anytime and anywhere, which has tremendous values for Smart Home users. The Web application is mainly responsible for reading sensors, storing readings, and monitoring home appliances over the internet. In the implementation, we use Google App Engine platform [13] for developing, deploying, and maintaining our Web applications because it is simple to use, scalable to service requests, and has built-in data store and a flexible interface.

Our Web applications is categorized into two main parts: a front-end and a back-end. The front-end serves as a Web client interacting with the user. The back-end serves as computing services for logic processing or storage services for data storing. We use HTML5 for describing web pages. HTML5 [14] allows a wider range of mobile devices to access our applications. We used JQuery Mobile [15] to support mobile interfaces and CSS (Cascading Styling sheets) for styling the web app interface. In addition, we use Ajax to create asynchronous web applications [16] [17]. The back-end services include measuring home conditions, monitoring home appliances, and controlling home access, etc. We also use storage as a service in the Cloud for hosting databases, objects, and message queue.

IV. USE CASE STUDIES

This section presents use cases, including measuring home conditions, managing appliances, and controlling and access.

A. *Measuring home conditions*

This use case develops the Cloud service for measuring home conditions, which is hosted on Google Cloud infra-structure. The measuring service allows the user to store data on home conditions and also to visualize them so that the user can view them anywhere, anytime. In this use case, we used the following sensor modules for measuring home conditions of temperature, humidity, ambient light and proximity. Fig. 4 shows the user interface of the measuring service.

- Proximity and ambient light sensor VCNL4000 [18]. It includes a signal processing IC and supports an easy to use I^2C bus communication interface. It allows to record the amount of ambient light in an area and also record the proximity of a given object to the sensor.
- Temperature-humidity sensor DHT22 [19] is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

B. *Managing home appliances*

This use case develops the Cloud service for managing home appliances which will be hosted on Google Cloud infrastructure. The managing service will allow the user to control the outputs of smart actuators associated with home appliances. Smart actuators are devices, such as valves and switches, which perform actions such as turning things on or off or making adjustments in an operational system. Actuators provide different functionalities, such as On/Off valve service, positioning to percentage open, modulating to control changes on flow conditions, emergency shutdown (ESD), etc. In this case, we used the Relay SPDT Sealed actuators [20] to provide On/Off functionalities to control house appliances such as lights, lamps and fans. To activate an actuators, a digital write command must be issued to the actuator. The following JSON message {“destination”: “3”, “source”: “1”, “commands” : {“4”: “H”}} instructs Arduino board 3 to set the pin number 4 to high, i.e., power on the associated actuator. Fig. 5 illustrates the user interface of the monitoring service.

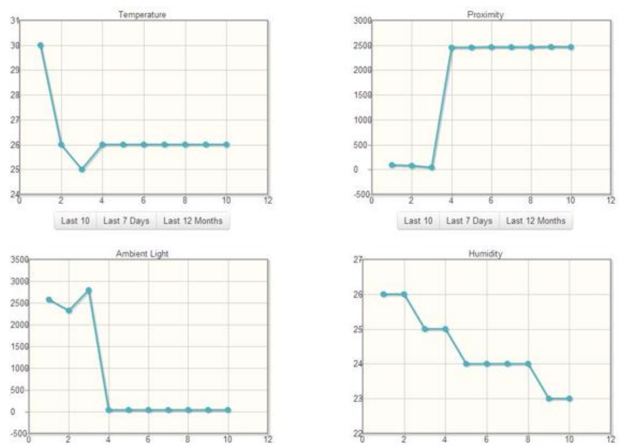

Fig. 4. User interface for measuring home conditions

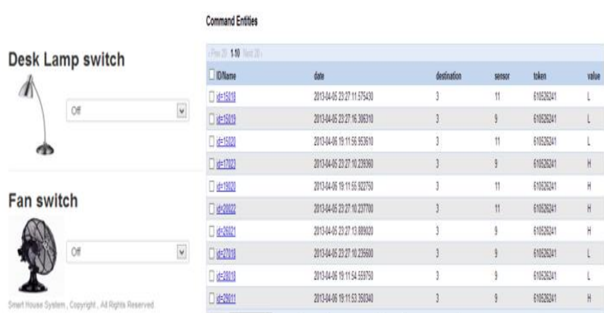

Fig.5. User interface for monitoring home appliances

C. *Controlling home access*

This use case develops the Cloud service for controlling home access which will be hosted on Google Cloud infrastructure. In the implementation, we use RFID tags -125 kHz and RFID reader ID-12 [21] to control home access. RFID tag -125kHz works in the 125kHz RF range with a unique 32-bit ID and not re-programmable. RFID reader ID-12 has a built in antenna with read ranges of 12+ cm. Every user of a home will have an RFID card associated with a unique ID. Before leaving or entering the house, the user will scan his/her card over the RFID reader next to the door. The scanned ID is sent from the reader to the associated Arduino board through serial protocol. The Arduino board posts the ID to the controlling service which compares the scanned ID against the authorized IDs in the database.

V. CONCLUSION

This paper explored the concept of Smart Home through integrating IoT with Web services and Cloud computing. Our approach consisted of embedding intelligence into sensors and actuators using Arduino platform, networking smart things using Zigbee technology, facilitating interactions with smart things using Cloud services for easy access in different locations, and improving data exchange efficiency using JSON notation. The approach was successfully used for demonstrating services for measuring home conditions, monitoring home appliances, and controlling home access. The infrastructure can be adopted for or adapted to other applications.

REFERENCES

[1] J.-Y. Son, et al, "Resource-aware smart home management system by constructing resource relation graph,” IEEE *Trans. on Consumer Electronics,* vol. 57, pp. 1112–1119, 2011.
[2] D.-M. Han and J.-H. Lim, "Design and implementation of smart home energy management systems based on zigbee,” *IEEE Trans. on* Consumer *Electronics,* vol. 56, pp. 1417–1425, 2010.
[3] C.-L. Wu and L.-C. Fu, "Design and Realization of a Framework for Human–System Interaction in Smart Homes,” *IEEE Trans. on* Systems*, Man and Cybernetics,* vol. 42, pp. 15–31, 2012.
[4] M. R. Alam, et al, "SPEED: An Inhabitant Activity Prediction Algorithm for Smart Homes,” *IEEE Trans. on* Systems*, Man and Cybernetics,* vol. 42, pp. 985–990, 2012.
[5] L. Chen, C. D. Nugent and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes,” *IEEE Trans. on Knowledge and Data Eng.,* pp. 961–974, 2012.
[6] Euler Proof Mechanism,“http://www.agfa.com/w3c/euler/, 2012”.
[7] Arduino, "http://www.arduino.cc/”.
[8] mbed, http://mbed.org.
[9] K. Gill, et al, "A zigbee-based home automation system," *IEEE Trans. on Consumer Electronics,* vol. 55, pp. 422–430, 2009.
[10] J. Han, H. Lee and K.-R. Park, "Remote-controllable and energy-saving room architecture based on ZigBee communication,” *IEEE Trans. on* Consumer *Electr. ,* pp. 264–268, 2009.
[11] Y. Doi, et al, "XML-less EXI with code generation for integration of embedded devices in web based systems," *Proc. of the 3rd Int’l Conf. on Internet of Things,* 2012, pp. 76–83.
[12] C. Severance, "Discovering JavaScript Object Notation,” *Computer,* vol. 45, pp. 6–8, 2012.
[13] A. Bedra, "Getting Started with Google App Engine and Clojure,” *IEEE Internet Computing,* vol. 14, pp. 85–88, 2010.
[14] “HTML5," *Http://www.w3. org/html/wg/drafts/html/master/.*
[15] "jQuery," *Http://jquerymobile. Com/,* accessed on4/19/2013.
[16] P. Mazzetti, et al, "Integration of REST style and AJAX technologies to build web applications; an example of framework for location-based-services," *Proc. of the 3rd Int’l Conf. on Info. & Communication Technologies,* 2008, pp. 1–6.
[17] J. Huang and B. Cheng, "Interactive visualization for 3D pipelines using Ajax3D," P*roc. of the Int’l Conf. on Networking and Digital Society,* 2009, pp. 21–24.
[18] V. Semiconductors, "Fully Integrated Proximity and Ambient Light Sensor with Infrared Emitter and I^2C Interface, "*http://www.Vishay.com/docs/83798/vcnl4000. pdf.*
[19] "DHT22 temperature-humidity sensor," *http://www. Adafruit. com/products/385,* accessed on 4/19/2013.
[20] "Relay SPDT Sealed - 20A,” https*://www. Sparkfun. com/datasheets/Components/T9A_DS.pdf.*
[21] "RFID Reader ID-12," *https://www. Sparkfun. com/datasheets/Sensors/ID-12-Datasheet.pdf.*