iToy: A LEGO-like Solution for Small Scale IoT Applications

Yi Ren, Muhammad Alfiansyah, Nyoto Arif Wibowo, Cheng-Wei Wu, JieFu Geng, and Yu-Chee Tseng
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Abstract—We can find various Internet of Things (IoT) products available in the market to accommodate users needs. Current IoT applications, however, built delicately for their usage with their own sensor devices and Apps. This limitation thus prevents users from handling flexible situations. In this paper, we propose iToy, an LEGO-like solution to integrate various off-the-shelf sensors for different IoT applications. In iToy, the off-the-shelf sensors are like LEGO plastic bricks, which can be assembled and connected to construct IoT applications. Any sensors constructed in an IoT application can be taken apart again and then used to make other IoT applications. We demonstrate through proof-of-concept prototype that iToy is user-friendly and suitable for forming different small scale IoT applications.

Index Terms—Internet of Things, Rule-based, Small Scale IoT Applications, App, Sensors

I. INTRODUCTION

Internet of Things (IoT) enables interaction between IoT devices, cloud server, and users in a way that we are able to closely connect the cyber world with the physical world. IoT applications have been widely deployed in smart home, e-health, intelligent cars for the sake of improving human living experience. Nowadays, the advance of smart phones, Arduino, Raspberry Pi, Edison, decreases the development time and cost for small scale IoT applications. Various IoT applications are emerging in the market, making our life much easier and smarter, e.g., health and fitness tracking [1], personal belonging tracker [2], [3], context aware marketing [4], [5], simple home automation [6]–[8], etc.

Those IoT devices and Apps, however, are designed for dedicated applications. Those IoT applications usually consist of a set of IoT devices and an intelligent controller (usually a smart phone). The IoT devices are used to collect data from users or environment, while the intelligent controller is used as User Interface (UI) to interact with users. For a user who brought three IoT applications, the user may get three sets of IoT devices and need to install three Apps in his smart phone. Due to conflict of interest, an IoT device (e.g., temperature sensor) from one application usually cannot be used in other applications, even if both are capable to handle the same task. It is a waste of money and time if a user cannot use those sensors interchangeably for different IoT applications or needs to install various Apps in his smartphone to manage the applications one by one. Therefore, a universal solution is desired, i.e., a single App can handle most of common tasks by using common sensors.

Inspired by LEGO, in this paper we propose iToy, a LEGO-like solution enabling users to create their own application using off-the-shelf sensors in the market. The off-the-shelf sensors are like LEGO plastic bricks, which can be assembled and connected to construct various IoT applications. Any sensors constructed in a IoT application can be taken apart again and then used to make other IoT applications. Specifically, the hardware components of iToy consists of an Intelligent Toy and several Dumb Toys (like LEGO plastic bricks). After setting trigger rules and actions (e.g., alarm, vibration, blink, etc.) in iToy App, the rule set and actions enable the Intelligent Toy to response actions according to the data sent from the Dumb Toy.

The contributions of this work are as follows. First, we propose and define iToy framework for the development of a LEGO-like universal solution for small scale IoT applications. Second, for prove-of-concept purpose, we demonstrate two IoT applications based on iToy with very simple rule settings. Given 5 min, even a 10 year old kid is able to form the IoT applications, showing the flexibility of the LEGO-like iToy design. Third, with the given IoT application examples, users can form their own IoT applications with a few simple steps. Allowing users with limited knowledge about sensors are able to generate a rule with various type of beacons.

II. iToy SYSTEM OVERVIEW

A. System Architecture

iToy system consists of three components: iToy App, Intelligent Toy, and Dumb Toy (Bluetooth 4.0 enabled sensors), as shown in Fig. 1 as an example. Generally speaking, both Intelligent Toy and Dumb Toy are devices with some sensors. Table I shows their features and differences. Dumb toy is a sensor node which is able to generate sensed data and send the data. Whereas, Intelligent Toy is more powerful with respect to three more features: Receive, Logic unit, and Action. With iToy, an IoT application can be formed in three ways:

- Intelligent Toy $\times 1$
- Intelligent Toy $\times 1 +$ Dumb Toy $\times m$
- Intelligent Toy $\times n +$ Dumb Toy $\times m$

B. iToy Hardware Components

1) Intelligent Toy: An Intelligent Toy is a smart phone or Embedded Device (e.g., Raspberry Pi, Edison, etc.) that can receive messages from Dumb Toy and perform pre-defined actions. Its main purpose is to implement users designed rules (store rules, Dumb Toy sensor data monitoring, and perform actions).

Table I: Characteristics of Intelligent Toy and Dumb Toy

Features	Dumb Toy	Intelligent Toy
Send Data	Yes	Yes
Receive Data	No	Yes
Logic Unit	No	Yes
Action	No	Yes
G-Sensor	Yes	Yes
M-Sensor	Yes	Yes
Gyroscope	Yes	Yes
ECG	Yes	Yes
GPS	Yes	Yes
Other Sensor	Yes	Yes
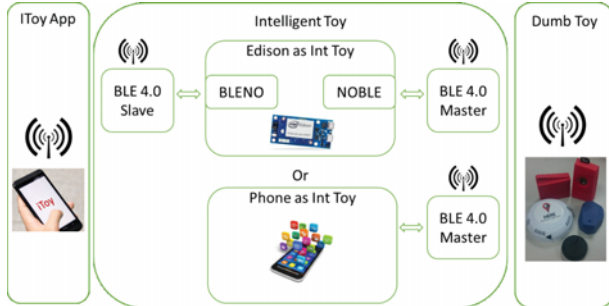
Figure 1: iToy system architecture.

- **Smart phone as Intelligent Toy.** Users can create rules by using iToy App in a smart phone. The smart phone then can perform user defined actions if the rules are hold.
- **Embedded devices as Intelligent Toy.** Users firstly create rules by using iToy App in their smart phone, and then send the rules to the embedded devices via BLE 4.0. Upon receiving the rules, the embedded devices will parse the rules, monitor Dumb Toy, and perform actions once the rules are hold.

2) Dumb Toy: A Dumb Toy is a sensor node or beacon with off-the-shelf sensors and BLE communication module. Dumb Toy will update its state or information to the Intelligent Toy periodically.

C. iToy App Software Design

We use sensor data from a sensor group to cooperatively trigger pre-defined actions.

1) iToy App: an App for creating rules for an IoT application: The main purpose of iToy App is to manage user designed rules, which are creation, editing, and deletion of rules. Rules are the embodiment of users will toward the environment condition represented by Dumb Toy sensor data values. For an example, user can set his phone to get notification when the temperature monitored by Dumb Toy-A surpass a certain value, or tell Intelligent Toy to take a picture when Accelerometer in Dumb Toy-B reach 2 G.

Each rule set consists of logic unit, association unit, exception unit, and action unit.

- Logic unit is a logic representation of a Dumb Toy, which consists of four items: Dumb Toy ID, the input value from
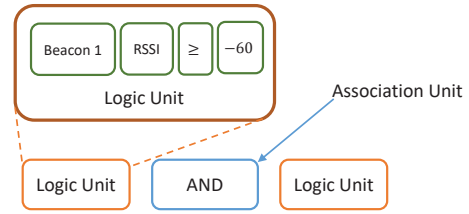

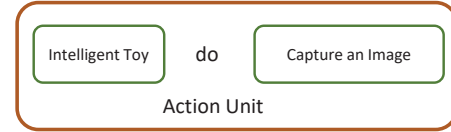Figure 2: An example of logic unit and association unit.

Figure 3: An example of action unit.

the Dumb Toy, logic operator (i.e., $=, \leq, \geq$, etc.), and a threshold value, see Fig. 2 as an example.
- Association unit is association operators, i.e., AND, OR, XOR, which is used to connect two or more logic units for complicated logic expressions.
- Action unit is a predefined action that a user wants the Intelligent Toy to do when all the logic unit combination sums up as True. An action unit consist of the Intelligent Toy ID and action as shown in Fig. 3. There could be many actions for a smart phone, e.g., alarm, vibrate, call someone, screen turn on, etc. Users can define the actions based on their needs.
- Exception unit is another logic unit that is designed to repel an event when it is true. Programmers can imagine it as an ordinary logic unit with Negation operator in front of it. Fig. 4 concludes the whole rules architecture and Fig. 7(c) illustrates its UI design.

D. Intelligent Toy Software Design

The software design of Intelligent Toy is presented in Fig. 5. Intelligent Toy is developed based on NodeJS environment, where BLENO and NOBLE NodeJS modules are used to handle Bluetooth 4.0 adapter. The Application will act as the center to control usage and modification of Rule Object. Rule Object in here is a set of rules that user had designed. It stores latest rules and process the incoming data from Dumb Toy if it

Intelligent Toy software consist of two major part: App Coordination Part and Monitoring Part. App Coordination Part is responsible to manage the synchronization between the Intelligent Toy and user iToy App. The synchronization information are User designed rules and Cloud API Key (e.g, Google Cloud Messaging API Key). For synchronizing user design rules, the App Coordination Part needs to handle user request to create, delete, or update rules, and manage notification settings. In Cloud API Key synchronization, Intelligent Toy needs to accept user's API key and store locally for later use.

Monitoring Part is responsible for searching, connecting, and monitoring users Bluetooth 4.0 Devices (Dumb Toys). Intelligent Toy then needs to connect to those devices and
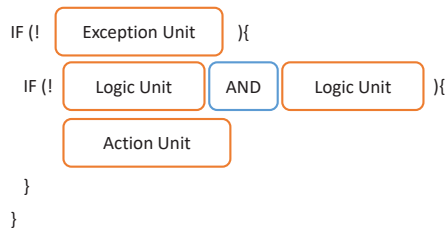
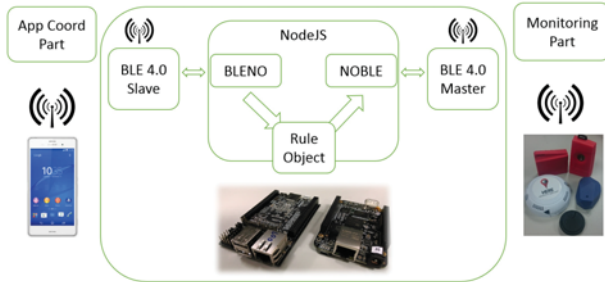Figure 4: iToy Apps rule expression.

Figure 5: Intelligent Toy software architecture

store its chosen sensor parameters continuously. If a user's designed rules need to include Dumb Toy's sensor data, then the Intelligent Toy needs to handle subscription to the corresponding data service and parameters as well as handle incoming data notifications. After all the needed data are gathered, then the Intelligent Toy needs to process it according to the user's rules. If certain event rule equals to True, then Intelligent Toy conducts the corresponding predefined actions (e.g, Notify users phone or Capture an image). It is worth to mention that the actions available in a Intelligent Toy depending on additional module it have. If it has Wi-Fi adapter, then it can connect to the cloud to notify the user or even store monitored data. If it is equipped with camera, then it can capture an image. Alarm or voice notifications can also be made just by installing a speaker.

III. PROTOTYPE & IMPLEMENTATION OF KOALA

We have built a proof-of-concept prototype of iToy for demonstration. A Samsung Galaxy smartphone running Android 4.44 is used as Intelligent Toy. The iToy App takes about 4.36 MB storage on the phone, and requires about 6.23 MB RAM for running. The Dumb Toy (called Koala) is shown in Figs. 6(a) and 6(b). It has accelerometer, gyroscope, magnetometer, and BLE communication module installed in it.

Here we use two examples, from simple usage to more complicated one, for the demonstration of iToy LEGO-like design. By setting a few rules and action parameters, one can build many IoT applications.

A. iToy as: Your Diary Peek Notifier

Let us consider a situation that a girl named Sophie concerned about the safety of her diary. She is afraid that someone in her house, probably her mon, :), trying to read her diary. How iToy can help to address this problem?

(a) Koala beacon used as Dumb Toy.

(b) Koala beacon inner component.

Figure 6: Dumb toy example.

Given one Intelligent Toy (iToy), two Dumb Toys (B1 and B2), and her smartphone (SP), we will show how the application is formed. The application has only one basic function: detect if the diary is read by someone (except Sophie herself). The system architecture is as shown in Fig. 7.

First, she needs to attach one dumb toy beacon (B1) on top of her diary, put intelligent toy (iToy) somewhere in the room and always bring the another dumb toy beacon (B2).

The next step is rule design phase. In the iToy App, she need to add a Logic unit which states if the gravity acceleration Z-axes from B1 passed certain value. Action Unit need to be created to command the intelligent toy to send notification to her phone. If a wireless IP Camera is available, then she could also include and set it up to take a picture. An Exception Unit then can be created to cancel all rules if beacon B2 is detected, which mean Sophie is around the area.

Shows in Fig. 7(a) whenever the dumb toy B1 detect changes in its sensor indicating someone is lifting or opening the diary, the intelligent toy will take a picture and send notification directly to Sophies Smartphone. Fig. 7(b) shows if Sophie is the one that is opening the diary then it will do nothing as long as dumb toy beacon B2 is around Intelligent Toy Coverage.
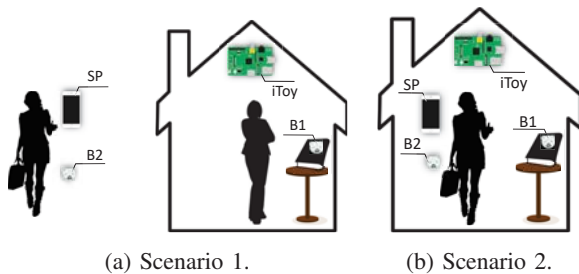
Rule creation for "Your Diary Peek Notifier" can be seen in Fig. 7(c).

B. iToy as: Life Bring Enhanced with Security Features

Considering a situation where a man named Andy always forgot some of his stuff. He decided to replicate LifeBring (lost item reminder), but adding another function which is to monitor if someone unauthorized is trying to open the door, an alarm will be triggered. LifeBring [8] is a personal belonging tracker, which take different approach to solve lost item problem compared to majority item tracker service in the market. It uses B-Door, a standalone observer which user put on their main door, to monitor any passing Beacon. If the user pass with one of the beacon missing then it will remind you that one or some of your stuff are left behind.
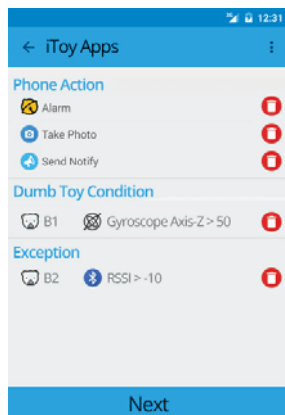
User need to set up a rule to help remind him if he forgot to bring stuff he has marked before he left home. First, user need to add an application named Life Bring in the iToy App interface. Then he need to mark any item he want to be notified if he forgot to bring by attaching dumb toy beacon on it. He also need to put intelligent toy on his main door.

Here, we show how to use iToy to design the functions of LifeBring. As shown in Fig. 8(a), an intelligent toy is attached
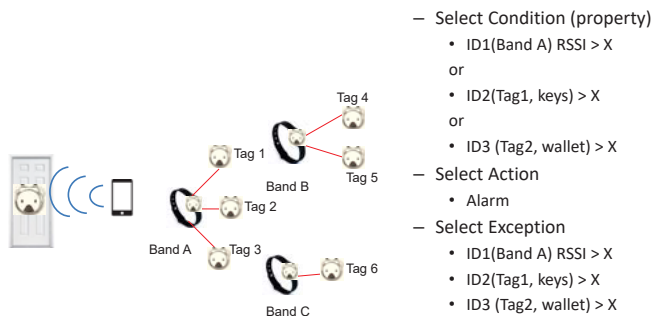
(a) Scenario 1.　　(b) Scenario 2.

(c) Rule created for Diary Notifier

Figure 7: Implementation of Diary Peek Notifier

(a) An example of LiftBring.　　(b) LifeBring rule setting.

Figure 8: Implementation of LifeBring.

something. To preserve the battery, exception can be made for the second beacon which can be in a smartwatch or smartphone object. When the second beacons signal strength is greater than 53dB, the Intelligent Toy will not trigger the alarm.

To apply this rule put the Intelligent Toy on the door. It will always monitor whether users forget to bring one or more key item or not when he pass through the door. To add the security enhancement, Andy have to create another application which will work in parallel with the previous one (LifeBring). In this matter, iToy support multiple App to run in parallel. In order to add simple security enhancement, Andy need to attach one more beacon sensor on the Door. Set Intelligent Toy to trigger the alarm whenever the beacons gyroscope sensor data pass certain value (Indication of someone opening the door) while beacons attached at the item is not present (Indicating that someone is not Andy). Same with previous example, Intelligent Toy can be used to take a picture if IP Camera is equipped.

IV. CONCLUSION

This is the first work addressing one application to manage mobile and on-the-spot IoT appliance, targeting multiple beacons in the process. iToy work with rule generation method to design an application for users' daily problems. In iToy, we introduce Intelligent Toy and Dumb Toy as the main part of iToy Application. The rule generation will be created in an App and then sent to Intelligent Toy. The iToy Apps will then control and manage connected Dumb Toys. iToy is well designed for individual users to tackle their daily problem, therefore both beginner and advance user can easily use it. To understand more how iToy work, we prepared a demo video which can be accessed at [9].

to the door and keeps sensing if there is any tags pass the door. There are three bands, Bands A, B, and C belonging to father, mother, and Andy. Each person has his personal belongings (e.g., key, phone, bags), so that Tags, 1, 2, ..., 6, are attached to those personal belongings. Also, those belongings are associated with a band. For example, Tags 1, 2, and 3 are associated with the Band A. So do the rest of tags. The rule settings are shown in Fig. 8(b). An action—Alarm—would be triggered if one of the conditions is not satisfied. That is, one tag is missing when a band pass the door, otherwise, the alarm will keep on silence.

Specifically, the rule need to be set up like this: For the beacon which need to be monitored, when the beacons signal strength is greater than 46dB then Intelligent Toy can be configured to trigger an alarm to notify user that he forgot

REFERENCES

[1] Y. J. Zheng and Z.Zhang, "Emerging wearable medical devices towards personalized healthcare." in *International Conference on Body Area Networks*, 2013, pp. pp. 427–431.

[2] R. S.Ahmad and Y.Zhang, "An empirical study of smartphone based tracking devices and analysis of its demand & market." in *International Journal of Economics Commerce and Management, vol II*, Dec. 2014.

[3] J. Stern. (2014, June 17) Finding the best lost-item trackers: Tile, trackr and duet reviewed. [Online]. Available: http://www.wsj.com/articles/finding-the-best-lost-item-trackers-tile-trackr-and-duet-reviewed-1403046981

[4] OnyxBeacon. (2015, Aug. 26) Onyxbeacon. [Online]. Available: http://www.onyxbeacon.com/solutions

[5] B. Levine. (2015, March 17) Smartfocus announces virtual beacons so retailers can draw in-store targeting. [Online]. Available: http://venturebeat.com/2015/03/17/smartfocus-announces-virtual-beacons-so-retailers-can-draw-in-store-targeting

[6] B. Ward. (2015, August 18) How the iphone 6s could control your home. [Online]. Available: http://www.techradar.com/news/digital-home/how-the-iphone-6s-could-control-your-home-1302107

[7] SmartThings. (2015, Aug 26) Smartthings. [Online]. Available: http://www.smartthings.com/how-it-works

[8] LifeBring. (2015) Lifebring. [Online]. Available: http://www.here-apps.com/lifebring.html

[9] M. Alfiansyah. (2015, Jun 2) itoy introduction video. [Online]. Available: https://www.youtube.com/watch?v=R3t5h-u-C6k