A Preliminary Component Model for IoT

Arbaayah Kamarudin¹, Ng Keng Yap¹, Mohamed Abdullahi Ali¹ and Lim Swee Phang¹

¹Department of Software Engineering and Information Systems, Faculty Computer Science and Information Technology, Universiti Putra Malaysia

gs49499@student.upm.edu.my, kengyap@upm.edu.my, mrmoo2030@gmail.com and www.limsp95@gmail.com

Abstract. The enthusiasm of IT entrepreneurs in producing Internet of Things (IoT) systems is undeniable as currently, the number of connected devices is enormously increasing. Many research has been done to efficiently develop IoT systems. IoT systems are usually engineered from scratch. IoT component models have been introduced but lack of generic development framework or model that supports high reusability and loose coupling in dealing with the heterogeneous devices that can hinder its development. Thus, an IoT component model is proposed. Meta-modelling has been used to define the component model where the specific interaction and composition standard in a component are abstracted. IoT component model is intended to develop a prototype for IoT development. With this IoT prototype, IoT system developers will not need to develop everything from scratch every time, as generic components can be reused even when it is applied in different domains or during system enhancement is required. Smart home IoT system has been selected as a case study to evaluate our prototype tool. In this study, we provide an alternative way to develop IoT software in component-based software engineering method. A prototype has also been developed to assist reusability and reduce coupling between modules.

1. Introduction
The exponential growth of connected objects that can participate in the Internet of Things (IoT) ecosystem is inevitable as nowadays we witness the tremendous progress of interconnection between digital and physical devices [2]. Examples of physical devices, such as sensors, lights, fans, watches, cameras, alarms, medical devices, transport devices, and any other numerous devices- are aimed to be connected over a network so that they can communicate with each other and make decisions intelligently, anytime and anywhere.

IoT components are IoT sub-systems that interact with each other in an IoT system and can be reused in a new environment [2]. It allows developers to develop the IoT systems from existing IoT components rather than from scratch. The reusability of IoT components benefits the developers and organizations in many ways such as to decrease bugs and defects in the systems, to shorten the development time and cost, and to increase productivity.

IoT component model is used to define IoT components and its standard of interaction and composition of IoT systems [3]. Similar to other software component models, the IoT component model uses port connector type architectural units as components, with method calls through port-to-port connections as the mechanism to compose components.
To develop IoT systems from scratch requires tremendous effort as developers must have an in-depth understanding of technologies and the new application domain where so many software components have to be engineered from scratch especially when the systems have to deal with a huge number of sensors and actuators [9].

IoT systems have been adopted by many companies adopting monitoring, fault detection, and diagnosis of manufacturing systems (MFDD) [7]. This hinders component reusability even though their goals are the same – to achieve monitoring, fault detection, and diagnosis goals. The similar issue has been addressed in [5] whereby no unified and interoperable system which could be reused and redeployed in future smart cities.

There is no generic development framework or model for such IoT systems, which cause the producing of application codes tangled with very specific scenarios or algorithms [5]. Lack of proper abstractions for application development is also one of the factors [5].

We found that most IoT systems apparently share 3 common component types, namely sensor, actuator, and controller. All of them are driven by software. Sensors and actuators, in particular, have their own common functions, i.e. sensors are used to read data from the environment; whilst actuators are used to actuate physical response to the environment, either mechanically or electronically. This phenomenon seems to hold for any kind of sensors and actuators. Although these devices may work in different modes, such as toggling, continuous, intermittent, we believe that they can be reused if they are made generic and highly configurable.

In this paper, we propose an alternative way to develop IoT systems that emphasize on reusability and loose coupling. This is achieved by defining an IoT component model with generic and highly configurable component types. We classify components into 3 main types in general, namely sensor, actuator, and software. The sensor component can be instantiated to drive any kind of hardware sensor, similarly, the actuator component can be instantiated to drive any actuator hardware. The software component can be instantiated to encapsulate any control logic. A set of these instances will be composed to build an IoT system.

Abstraction and generalization are keys at large in defining the IoT component model. For instance, the sensor component is the type that generalizes the data reading function for all kinds of sensors. The data reading mode and data type are the intrinsic attributes of the sensor component, in which we believe they should be configured in their respective instance to suit the underlying sensor hardware and application purpose. For example, for temperature sensor hardware, the sensor (software) component instance can be configured to read a decimal value from the sensor hardware every second in one application; or read an integer value every 10 minutes in another application. The underlying sensor hardware may also vary even for the same configuration.

This paper proposes an IoT component model that considers reusability and loose coupling. Metamodelling was used to define the component model. The IoT component model acts as the basis to develop a modelling tool prototype for IoT system. With this tool in place, when the performing system enhancement or developing a new system in different application domains, IoT system developers can reuse the components instead of developing it from scratch.

The remainder of the paper is structured as follows: Section 2 reviews existing research and related work. Section 3 presents the proposed IoT component model. Section 4 demonstrates the development of the IoT prototype tool. Section 5 discusses a case study. Section 6 presents discussion. Finally, Section 7 concludes our research.

2. Literature
Table 1 compares literature related to IoT architecture and determines whether software component models were defined or applied in these researches. We found that most of the literature suggests IoT system development without any software component model defined, there are prone to the adoption of SOA, Cloud API, or web services. Although [12] suggests component-based development, the authors did not generalize components like what we propose herein. Therefore, the component model is not IoT specific.
### Table 1. Comparison for various IoT Architecture

| Papers | Type of architecture                                                                 | Component model                                                                 |
|--------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| [4]    | 4 stages of IoT - Stage 1, where sensors, actuators, or any devices produce, receive and process the signals collected, Stage 2 where the data collected from the sensors aggregated and converted into digital stream, Stage 3 where the digitized data is being processed and Stage 4 where the data is forwarded to physical data centres or cloud-based systems. | No component model suggested                                                        |
| [16]   | Integrated platform for heterogeneous systems which consists of IoT middleware and information bus. Heterogeneous systems consist of sensors to collect data to be supplied to the middleware. | No component models suggested systems as the resources of the data.               |
| [13]   | Layered architecture consists of 4 major layers. Smart Objects layer contains sensors and actuators, Networking Infrastructure layer for SaaS to intermediate smart objects with Information Service layer. Application layer supports various application such as security, home automation, and energy. | No component models suggested. Smart Objects layer contains sensors and actuators but not encapsulated in components. |
| [12]   | The sensor node software architecture for this framework is designed in a multi-layered approach, which is the Application Layer (APL) and the Operating System Abstraction Layer (OAL). Components in each layer can be combined as one unit of concrete sensor node’s software architecture. | There are component models but not specifically for sensors and actuators. Sensors and actuators as nodes. |
| [15]   | Sensors are represented as nodes which designed in a distributed architecture.        | No component models suggested.                                                    |
| [11]   | Core microservice to control and coordinate other microservices. Device microservice contains the sensors and actuators. | No component models suggested.                                                    |
| [8]    | The framework basically consists of sensors and data collection area, cloud-based backend, domain-specific software components and applications, and services area. | No component models suggested.                                                    |
| [3]    | Semantic engine applied to IoT and smart cities, where they designed an entire workflow/semantic engine to semantically interpret and reason over data by providing services hiding semantic web technologies to IoT developers and end-users. | No component models suggested, only the semantics which limited to smart cities.   |

The study in [14] identifies top 10 industries investing in sensors, which are Energy and Mining, Power and Utilities, Automotive, Industrial, Hospitality, Healthcare, Retail, Entertainment, Technology, and Financial Services. These are the most affected areas by the expeditious dispersion of IoT technology, yielding numerous sensors to be developed.

Smart city is one of the examples of IoT application that society dreams of. For instances, smart surveillance, smart alarm, automated transportation, smart energy management systems, water distribution are IoT applications for smart cities [1]. [1] proposed a unified semantic engine for IoT and smart cities that they analyzed from web technologies implemented in IoT systems. However, the components of the semantic engine are difficult to be reused in other application domains as they are very specific to smart cities.

Apart from that, a generic microservice system framework for IoT application has been proposed in [10] that designed with a core service to control and coordinate other microservices. However, there is no component model adopted in this architecture to support reusability of the microservices.
3. Proposed Component Model

A Component model for IoT system has been proposed in this paper where the key component types are sensor and actuator. The component model is defined in meta-model as in Figure 1. This IoT component model defines generic IoT components. It also defines appropriate generic elements, composition rules to build an IoT system. In IoT systems, generic component types are sensor, actuator, and software.

In the meta-model, Component is the main class in the proposed component model, Sensor, Actuator, and Software are inherited from the component. A Component should have required ports, provided ports and may have CU (Computation Unit). A provided port has reference to required port, controllers and parameters when the component is composed with other components. A controller can be of types: branching, sequence, and iteration. An Iteration has a For and Break elements. A controller has provided ports, and required ports. A for has a CounterLoop, an IterativeLoop, and Condition elements. A branching can be of type: Switch or If: A Switch has a Break and a Case. An If has a Condition. A software component should have provided ports and required ports. However, a sensor and actuator should have a provided port with required port absent.

![Figure 1. Meta-Model on Proposed IoT Component Model](image)

4. Development IoT Tool

This IoT system modelling tool prototype is developed with Obeo Designer software. Obeo Designer is a graphical modelling workbench; we used it to facilitate the development of this tool.

The GUI of the development tool shown in Figure 2 provides component drag and drop to reduce the developer’s effort to do coding. The palette section on the right of the screen consists of a list of components to create IoT component instances, its port, and associations. The components can be dragged from the palette and dropped into the canvas in the middle of the screen.
5. Case Study

A smart home system [17] has been selected as a case study to evaluate the proposed IoT component model and its development tool. Two quality attributes were measured in this evaluation which namely: reusability of the components and loose coupling between components.

The selected case study allows us to describe the functions of the system and how it was implemented from scratch. The system presented is for monitoring and controlling smart home environment. As mentioned in [17], a smart home referred to a Internet-connected home or eHome as an environment for living that has highly advanced automatic systems.

The proposed smart home system mentioned in [17] consists of applications built on top of an IoT infrastructure. The main functions are:

5.1. Alert
The smart home system is able to sense the environment and sent alerts to users accordingly. The alert contains information related to the environmental data, which may include the levels of different gases in the environment, temperature, humidity, light, and intensity. An alert may be sent by email, text message, tweets or any social media on a regular basis at a predefined time.

5.2. Monitor
The smart home system is capable of monitoring its surrounding using sensors in general. It keeps track of every activity in a smart home where any further action can be taken or a decision to be made accordingly.

5.3. Control
The smart home system allows user to control different activities, which may include switching on and off the lights, air conditioners and other appliances, lock and unlock the doors, open or close the windows and doors and so on. They are actuators in general.
5.4. Intelligence

The smart home system can automatically make a decision upon the occurrence of various events. It can identify and react according to changing conditions and events. This is software-driven in general.

Figure 3 shows how Smart home system [17] was developed using FLIP. FLIP [17] is an open-source IoT platform, which has been used to develop the smart home system in the case study, which requires a lot of coding knowledge, in this case, C and Python.

Developers need to learn the programming skill to write codes. If the codes are written by experienced developers, it may be difficult to reuse as the codes are very system-specific.

This paper shows an alternative way to develop the smart home system based on the proposed IoT prototype tool as in Figure 4. The smart home system gathers sensor data from the environment and responds back to the environment through actuator according to the rules that have been set.

In Figure 4, boxes in blue represent sensor component, red represent software components and green represent actuator. The smart home software component is the main software component with three sensor types namely: light sensor, PIR sensor, temperature, and humidity sensor, which will gather sensor data from the environment.

The actuator called lamp will perform on/off tasks. The intelligent component is a software component that contains logics, which will automatically make decisions upon the occurrence of various events. The alert component is another software component, which is able to send alerts to the user.
accordingly. As the components are independent of each other, the coupling between components is reduced. This promotes the reusability of the components.

Generic sensor and actuator components can be configured, and incrementally composed to the system. As the system evolves, components can also be replaced to update or upgrade the system. The generic sensor and actuator components are highly reusable.

6. Discussion
We do not claim that our approach is a silver bullet to IoT system development. Nevertheless, it does provide an alternative practical way for building IoT systems. We advocate that common component types should be generalized for reuse. This idea has been duly implemented as the IoT component model and subsequently the modelling tool prototype. We also demonstrated how these generic components can be reused and configured according to the IoT system requirements using a smart home case study.

Our modelling tool prototype provides an intuitive visual modelling view (refer Figure 4) which will inherently reduce the learning curve for developers. However, the modelling tool prototype is still under development. The component model is also in its preliminary stage and requires further refinement. Upon completion, the code generator is expected to generate code to compose components according to the system design model. This will significantly reduce IoT system development cost and time, and subsequently increase productivity.

Currently, most of the IoT systems are developed by coding and calling the web services as programming API. Web services provide essential distributed computing power on the cloud to complement the computing power on edge devices [8]. The typical example of IoT systems that require intensive central data processing from distributed edge devices is weather forecast. In our component model, web services can be implemented and encapsulated as the Computation Unit (CU) in our generic component.

7. Conclusion
In this paper, we have proposed an IoT component model to generalize common component types across different IoT systems, namely sensor, actuator, and software. The component model was defined using the meta-modelling technique. Then, we implemented a component-based software modelling tool according to the component model to evaluate how practical the component model is.

Although the component model is in its preliminary stage, whereby the prototype is only capable of modelling systems with a subset of controllers and the code generator is still incomplete, we could boldly conclude that the prototype has improved the productivity of IoT system development, at least in system design.

System design can be done using the drag-and-drop feature provided by the modelling tool prototype. This has also been demonstrated using a smart home system. The use of generic reusable and configurable components for sensors and actuators renders the IoT systems built using this model loosely coupled.

We have demonstrated that system design can be achieved by the drag-and-drop feature provided by the modelling tool prototype with a smart home system case study. The use of generic reusable and configurable components for sensors and actuators has also rendered the smart home system loosely coupled.

8. Acknowledgement
The authors of this work gratefully acknowledge Ministry of Education (MOE), Malaysia for their financial support under Fundamental Research Grant Scheme (FRGS) (Ref:FRGS/1/2015/ICT01/UPM/02/6).

References
[1] A. Gyrard and M. Serrano 2015 A Unified Semantic Engine for Internet of Things and Smart Cities: From Sensor Data to End-Users Applications IEEE International Conference on Data Science and Data Intensive Systems Sydney, NSW pp. 718-725.

[2] Ayala, I., Amor, M., Fuentes, L., & Troya, J.M 2015 A Software Product Line Process to Develop Agents for the IOT Sensors Pg 15640-15660.

[3] B. Councill, G. T. Heineman Definition of a Software Component and Its Elements Available from: http://heim.ifi.uio.no/~frank/inf5040/CBSE/Component-Based_Software_Engineering_-_ch1.pdf.

[4] Bharani 2019 What is IOT (Internet of Things)? IOT Architecture Explained Available from: https://www.edureka.co/blog/what-is-iot/ Accessed 9-10-2019.

[5] C.M. de Farias, I.C. Brito, L. Pirmez, F.C. Delicato, P.F. Pires, T.C. Rodrigues, I.L. dos Santos, L.F.R.C. Carma, T. Batista 2017 COMFIT: A development environment for the internet of things Future Generation Computer Systems. 75 128–144.

[6] Flip Smart Home Kit n.d. Retrieved from http://myflip.io/Pages/SmartHome.html.

[7] I. Yen, S. Zhang, F. Bastani and Y. Zhang 2017 A Framework for IOT-Based Monitoring and Diagnosis of Manufacturing Systems 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE) San Francisco CA pp. 1-8.

[8] Im, J., Kim, S., & Kim, D 2013 IoT mashup as a service: cloud-based mashup service for the Internet of things. In 2013 IEEE International Conference on Services Computing pp. 462-469.

[9] Kamienski, C., Jentsch, M., Eisenhauer, M., Kiljander, J., Ferrera, E., Rosengren, P., Thestrup, J., Souto, E., Andrade, W.S., & Sadok, D.F 2017 Application development for the Internet of Things: A context-aware mixed criticality systems development platform Computer Communications 104 1-16.

[10] L. Sun, Y. Li and R. A. Memon 2017 An open IOT framework based on microservices architecture China Communications vol. 14, no. 2, pp. 154-162.

[11] Lengyel, L., Ekler, P., Ujj, T., Balogh, T., & Charaf, H 2015 SensorHUB: an IOT Driver Framework for Supporting Sensor Networks and Data Analysis International Journal of Distributed Sensor Networks, 109.

[12] Nguyen, X. T., Tran, H. T., Baraki, H., & Geihs, K 2015 FRASAD: A Framework for Model-driven IOT Application Development IEEE 2nd World Forum on Internet of Things (WF-IOT) 387 – 392.

[13] R. G. Anvekar and R. M. Banakar 2017 IOT application development: Home security system 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development Chennai pp. 68-72.

[14] S. Bauer, F. Burkitt, C. Curran, L. Gioia 2014 Sensing the Future of The Internet of Things, PwC 6th Annual Digital IQ.

[15] S. Siddula, P. Babu and P. C. Jain 2018 Water Level Monitoring and Management of Dams using IOT 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages Bhimtal pp. 1-5.

[16] T. Joseph, R. Jenu, A. K. Assis, V. A. S. Kumar, P. M. Sasi and G. Alexander 2017 IOT middleware for smart city: An integrated and centrally managed IOT middleware for smart city 2017 IEEE Region 10 Symposium Cochin pp. 1-5.

[17] T. Malche, P. Maheshwary 2017 Internet of Things (IoT) For Building Smart Home System International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) 65-70.