A Framework for a Multisensory IoT System Based on Service-oriented Architecture

Sevendi Eldrige Rifki Poluan* and Yan-Ann Chen*†

*Dept. of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan

†Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taoyuan, Taiwan

E-mail: se7en.thee@gmail.com, chenya@saturn.yzu.edu.tw

***Abstract*—Multisensory IoT system contains multiple types of sensors, actuators, and computation devices. In this kind of system, providing different viewpoints of observations by heterogenous sensors such as RGB camera and wearable sensors can improve the performance of intelligent systems. The issues of building this system are that we have many components running on different platforms and implementing with different programming languages. It causes the complexity of architecture design, interoperability of components, and portability and also limits the system scalability. Therefore, we propose a reference design which is based on service-oriented architecture for alleviating these issues. We also present a case study of system prototyping which is constructed by adopting an RGB camera and smart insoles to create an identity recognition system.**

***Index Terms*—Cyber-physical System, Heterogeneous Network Applications, Internet of Things, RESTful API, Multisensory Service-oriented Architecture, Transmission Management System, Web Services.**

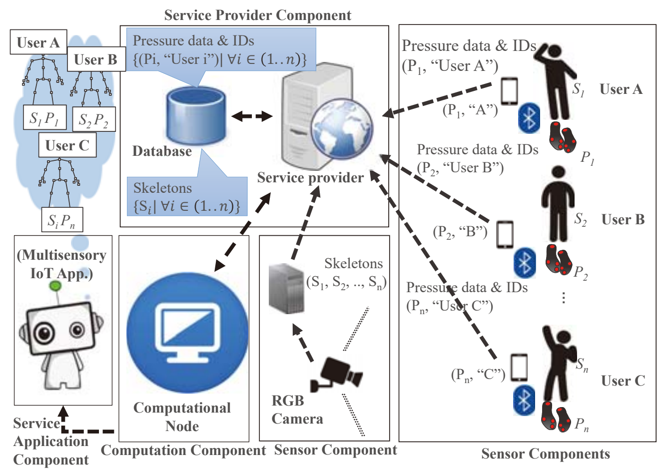

Fig. 1. System scenario

I. Introduction

Nowadays, with the explosive growth of *Internet of Things (IoT)* technologies, industry and academia are exploring the possibilities of integrating intelligence into IoT for building smart city, smart environment, and smart industry, etc. Some IoT systems [1][2] may only utilize single sensor modality for inference or recognition. However, multimodal approaches for intelligence systems [3][4][5][6][7] are prevailing techniques for enhancing the recognition or prediction accuracy. They collect multisensory information from heterogeneous sensors, such as RGB camera and wearable accelerometer, to observe the physical world with different viewpoints. Therefore, these applications can obtain more information from these viewpoints. When building this kind of multisensory IoT application, there may exist multiple sensing, actuating, and computation devices which runs on different platforms. But, the major challenges of these systems are the complexities of architecture, system integration, system migration, and task management. Thus, we present a reference framework for a multisensory IoT system which provides scalability, interoperability, and portability.

For instance, Amazon Go [3] , which is one of the popular unmmaned stores, is an multisensory IoT application. It provides the “Just Walk Out” shopping experience where consumers grab products from shelves, directly put them into their shopping bags, and automatically check out without the cashier. For fulfilling this system, we have to deploy sensors to obtain the status of products, to recognize behaviors of consumers, and to detect these products are took by whom. The collected sensor information may be processed on the edge device, proprietary server, or on the cloud. So do the computations of behavior recognition and detection. Therefore, we have to modularize the software components such that we can easily deploy the programs onto different platforms and exchange sensory information among programs without modifying the software architecture. To achieve this purpose, RESTful-based *Service Oriented Architecture (SOA)* can facilitate this system for its ability of communication and integration of software components or external applications.

REpresentational State Transfer (REST) is a software architectural style for retrieving and delivering resources in terms of the resource identifier with stateless operations [8]. Due to its lightweight operations, low bandwidth usages, and adaptable data formats, REST style becomes popular to develop web services. RESTful system refers the system which uses REST paradigm. Usually, the implementation of RESTful system is based on HTTP which has the cross-platform capability. It can be used for exchanging data between applications or systems without being limited to a specific system. Also, it supports inter-process communications, such as communications between Java and Python programs, or Windows and Linux applications. Therefore, we consider integrating this style into our system.

Fig. 1 shows the scenario how our framework works. We utilize a multisensory IoT application for identity recognition [7] as the case study. In the system architecture, we split this system into multiple software components, namely sensor components, service provider component, computation component, and service application component. Specifically, our sensor components are divided into two independent components because of the multisensory information. The first kind of sensor component is an RGB camera to capture images and extract the information of body skeleton $\{S_1, S_2, \ldots, S_n\}$ from users. The second kind of sensor component is a pair of smart insoles to collect pressure readings $\{(P_1, \text{"}A\text{"}), (P_2, \text{"}B\text{"}), \ldots, (P_n, \text{"}C\text{"})\}$ from each user. Here, A, B, and C refer to the identities (IDs) who wear the correspoding smart insoles. Next, the computation component is to recognize the corresponding ID by fusing multisensory data when providing a set of skeleton data and pressure readings. The number of sensor components may vary according to the users needs. The communication protocol is to communicate the service provider component with the insoles via Bluetooth and with RGB camera via a wired cable. Smartphone device and computer act as gateways to deliver sensory data to service provider component via REST API for data storage.

The main contributions of this work are as follows. First, we define a reference framework for the collection of multisensory data where the programs can run on cloud system and edge devices for portability. Second, we investigate the possibility of applying RESTful-based Service Object Architecture (SOA) for multimodel IoT applications for interoperability.

The remainder of this paper is organized as follows. In the Section II, we will discuss about the overview of our proposed framework. Section III shows the system features and components. Finally, Section IV concludes this paper.

II. Framework Overview

Given some sets of the collected sensor data, the framework we propose will try to discover some patterns from the data in order to be considered as features. The extracted information then will facilitate the works of multisensory IoT applications to do its tasks. Fig. 3 outlines our proposed service-oriented architecture framework for this multisensory IoT system. This framework comprises with four layers: physical layer, gateway/middleware layer, and service integration layer. Generally, each layer has different components and functionalities. The details explanation about each component will be described in the following section. In this part we will discuss briefly about the functionality of each layer and its relationship between one to another.

The physical layer contains a number of sensor devices. This is the layer of which the data are collected from the environment. The communication layer is a protocol to the sensor devices to deliver their sensory data to the middleware layer. Finally, through RESTful API, it will forward the data to the cloud services in the service integration layer. On that layer, there are three independent components. They are service application component, computation component, and service provider component. We connect service application component to computation node to get the computation results generated by the computation node. The computation node itself communicates with service provider component to collect the multisensory data sent from the middleware layer. We fulfill the communication by web service methodologies. Middleware layer is built as a gateway to the physical layer to deliver their data to the service integration layer through communication layer. It can be smartphone, edge node, Raspberry Pi, TX2, etc, or any other node. Whereas, to the communication layer it can be by Bluetooth, WiFi, wired, ZigBee or any other communication protocols.

A. RESTful Web Services

Fig. 4 describes how RESTful provider integrates the multisensory IoT applications with the four different client applications. We show them in the Fig. 1. We assume that each client has different application environments. In response to this, we explorer the communication between Android, iPhone, Linux, and Windows client applications to complete this purpose. Hence, the use of web service opens up opportunity to this kind of exchange data communication to be accomplished. Web services grant access to multiple dependent/stand-alone applications and environments to share resources and services. Their provided access is identified by clients via uniform resource identifiers (URIs) using standardized interface and protocol, typically HTTP. URIs are a based host name of which clients can identify particular resources of the service provider [9]. Whereas to use the resources, there are three HTTP methods that we use for the communication, namely POST, GET, and DELETE methods [10]. Those things are the application programming interfaces (APIs) between our stand-alone client applications and RESTful provider to have an interaction.

B. Service Oriented Architecture (SOA)

We follow the three majors roles of building SOA model in our system, that is discovery agencies, service provider, and service requester [11]. Fig. 2 shows the workflow of these three roles. Discovery agency or service registry contains information or description of the service provider. All the information of storing, deleting, and updating information are stored in the service registry including methods and resources to compute and perform users identification and recognition. Service registry acts like a broker to any potential client request to use or access resources from the service provider. When clients want to communicate or interact with the service provider, it will contact the discovery agency to provide information and ways to use the services provided by the service provider. Service provider is the one that grants and provides access to the outside applications and allows them to use and access its exposed resources through APIs. Whereas, the service requester is the application clients that give request to the service provider to use their services or to publish some data.

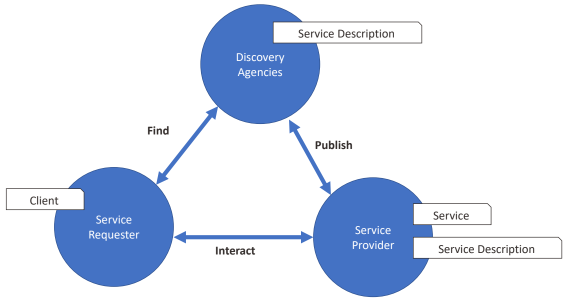

Fig. 2. Service Oriented Architecture (SOA)

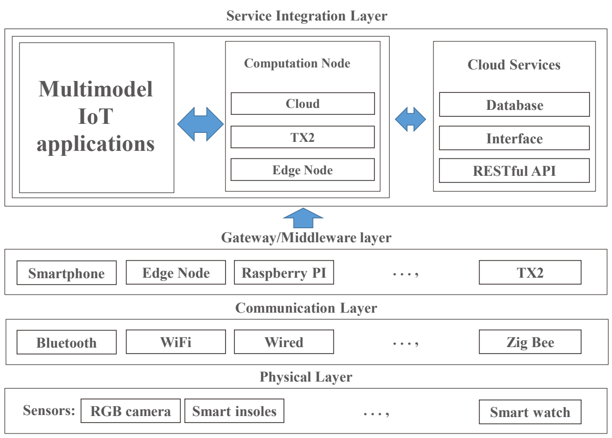

Fig. 3. Proposed framework

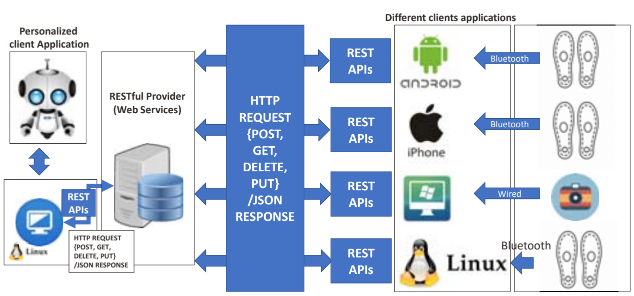

Fig. 4. Multi-environment services

III. Feature Components

Our purpose is to build a framework for multisensory IoT applications using RESTful-based SOA. In the following, we describe the feature components of this framework.

We divide this framework into four different components. They are (a) sensor components, (b) computation component, (c) service application component, and (d) service provider component. Those features of all four components are discussed as follows.

A. Sensor components

We consider to equip this system with a number of sensor components for capturing events or changes in environment.

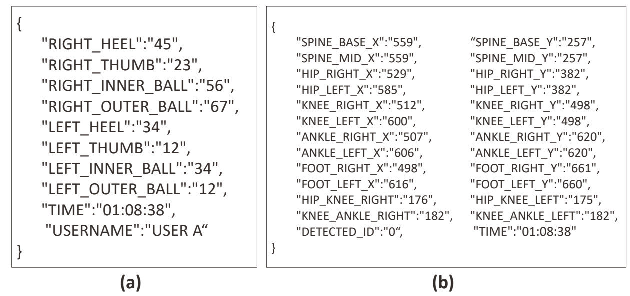

Fig. 5. Sample JSON format of (a) pressure (b) skeleton data.

The number of sensors may vary based on the users needs. Specifically, each sensor device collects their data with identification number. Those data then are sent in real-time to the gateway device via a communication protocol. At the gateway, a module is provided to handle and format the data into a JSON format and forward them to the cloud services via RESTful API interface. Fig. 5 shows a sample JSON data format of skeleton and pressure data. For further computation the data will be used.

B. Computation component

There are five main functions of the computation component that we provide. Firstly, to retrieve all the sensor data from service provider. Secondly, to handle all the computation tasks, e.g., feature extraction, data fusion, classification, etc. Thirdly, to store the final results into its local database. Fourthly, to provide services to the multimodel IoT applications in getting the final results. Lastly, to send request to the service provider to delete all the data that have been done to use and compute. In this case, the last function is optional. It depends to the users needs.

C. Service application component

This component is the end point of the system. This where the multisensory IoT application is. The works of sensor components, service provider component, and computation component are to be done for the sake of this component. One major module of this component is to synchronically request access to the computation component if there are new updated results to be returned. The functionality of this component is based on what purposes it is created.

D. Service provider component

On the cloud server, we define a web service with a database to provide an end-user interface. Each client interacts with the server using REST API through HTTP protocol. We may say that this is a service center of clients to be integrated one to another. It does not matter what environments the clients are built and what programming languages are used to deploy their applications. Fig. 4 visualizes the relationship among different client environments and the service provider. Each client has been organized according to their tasks and functionalities. Moreover, not all the clients can have the same authority to have access to the resources provided by the web service. We use JSON web token (JWT) authentication to define a compact and self-contained way for our system to securely ensure the data integrity of both sender and receiver [12]. There are there main HTTP methods, POST, GET, and DELETE, where we define for the system integration between client and the service provider. The allocation of each method to the client sides is described as follows. The gateways of sensor devices use POST method to post sensor data to the database server. The sensor data then are requested by the computation component using GET method. Once the computation is done, it sends a request to the service provider for the deletion to remove the used data where this function is optional according to the users needs.

IV. Conclusions

In this paper, we proposed a reference design which is based on service-oriented architecture for multisensory IoT systems. We presented a case study by adopting a RGB camera and smart insoles for an identification system. We used RESTful web service to implement this work which allows this system to have no restriction on platform or programming language that are used. This system is centralized into the service provider node. Another node is provided for the computation handling. It handled all related computation, calculation, classification, and identification. This computation node not only can be deployed on the cloud, but also can run on any edge devices, e.g., computer desktop, TX2, etc. Through that way, it allowed our system to easily migrate without affecting the existing components. Our application of the case study showed an opportunity of this framework to work for multisensory IoT systems.

Acknowledgement

Y.-A. Chen's research is co-sponsored by MOST 107-2218-E-155-007-MY3, Taiwan and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taiwan.

References

[1] Y.-A. Chen, J. Chen, and Y.-C. Tseng, "Inference of Conversation Partners by Cooperative Acoustic Sensing in Smartphone Networks," *IEEE Trans. on Mobile Computing*, vol. 15, no. 6, pp. 1387–1400, 2015.

[2] T.-H. Chiang, Y.-A. Chen, C.-T. Chang, L.-J. Chen, and Y.-C. Tseng, "Appliance Control by TDoA-based Localization and Gesture Recognition," in *Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018.

[3] A. Polacco and K. Backes, "The Amazon Go Concept: Implications, Applications, and Sustainability," *Journal of Business and Management*, vol. 24, no. 1, pp. 79–92, 2018.

[4] H. Li, P. Zhang, S. A. Moubayed, S. Patel, and A. P. Sample, "ID-Match: A Hybrid Computer Vision and RFID System for Recognizing Individuals in Groups," in *Proc. of ACM CHI Conference on Human Factors in Computing Systems (CHI)*, 2016.

[5] D. Mungtavesinsuk, Y.-A. Chen, C.-W. Wu, E. Bajo, H.-W. Kao, and Y.-C. Tseng, "Using Nonverbal Information for Conversation Partners Inference by Wearable Devices," in *Proc of International Conference on Internet of Things as a Service (IoTaaS)*, 2017.

[6] W.-C. Chang, C.-W. Wu, Y.-C. Tsai, C.-J. Lin, and Y.-C. Tseng, "Eye On You: Fusing Gesture Data from Video and Inertial Sensors For Person Identification," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[7] S. E. R. Poluan and Y.-A. Chen, "Using Smart Insoles and RGB Camera for Identifying Stationary Human Targets," in *Proc. of IEEE International Conference on Awareness Science and Technology (iCAST)*, 2019.

[8] C. Pautasso, "RESTful Web Services: Principles, Patterns, Emerging Technologies," in *Web Services Foundations*. Springer, 2014, pp. 31–51.

[9] B. Dingwall, S. Vlasenko, and M. Nezarati, "Method and system for a uniform resource identifier URI broker," Oct. 15 2019, US Patent 10 447 797.

[10] R. Lucchi, M. Millot, and C. Elfers, "Resource oriented architecture and rest," 2008, technical Report, European Commission Joint Research Center.

[11] S. Kumari and S. K. Rath, "Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration," in *Proc. of IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015.

[12] S. Ahmed and Q. Mahmood, "An Authentication based Scheme for Applications Using JSON Web Token," in *Proc. of IEEE International Multitopic Conference (INMIC)*, 2019.