

Distributed rule execution mechanism in smart home system

Agung Setia Budi, Hurriyatul Fitriyah, Eko Setiawan, Rakhmadhany Primananda, Rizal Maulana

Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia

Article Info

Article history:

Received Mar 22, 2021

Revised Mar 30, 2022

Accepted Apr 12, 2022

Keywords:

Centralized

Distributed

Fault-tolerance

Master station

Smart home

ABSTRACT

Smart home systems become an interesting topics in the last few years. Many researchers have been studied some features. Most of smart home system use a centralized architecture know as centralized smart home system (CSHS). The centralized mechanism is easy to manage and to configure. However, in fault-tolerant system paradigm it produces a problem. The entire system will fail, if the master station fails. Another problem of CSHS is centralized mechanism gives more data-flow. This condition makes the system has a greater delay time. To solve the problem, we propose a distributed rule execution mechanism (DREM). Compared to the centralized mechanism, the DREM allows a device to provide its service without any commands from the master station. In this mechanism, since the information does not need to go to the master station, the data-flow and the delay-time can be decreased. The experiment results show that the DREM is able to mask the failure in the master station by directly transmit the data from trigger device to service device. This mechanism makes the services provision without master station possible. The mathematical analysis also shows that the delay time of the service provision of the DREM is less than the delay time of CSHS.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Agung Setia Budi

Faculty of Compter Science, Universitas Brawijaya

Veteran Street, Number 8, Malang 665145, Indonesia

Email: agungsetiabudi@ub.ac.id

1. INTRODUCTION

Smart home system is a living environment which is equipped with smart advanced technologies that can predict and give respond to the requirements of home residents [1]. For the last few years, smart home researches have been increasing drastically. Many researchers have been studied some features of smart home system such as activity recognition [2]-[4], appliance control system [5]-[7], home security [8], [9], automation system based on internet of things (IoT) [10], [11], energy consumption monitoring [12], and balanced energy usage [13].

The technologies advancement of the sensor, actuator, signal processing, computation, and communication makes the smart home system possible to have all of its features. These technologies are used and connected cooperatively in a specific architecture and run by some algorithms. The sensors that are placed at the inside and outside of the house acquire the information for the smart home system. The signal processor then optionally process the sensor data and then transmit the processed data using wire or wireless communication technology to the master station. The data then are processed by the master station. The data processing are based on the knowledge base which includes reasoning machine and user preference to perform the best

services for the home residents.

Some approaches are proposed to provide the right service at the right time in an effective and efficient way for the home residents such as multi-stage scheduling [14], context-aware scheduling [15], [16], and centralized framework [17]. All of these approaches proposed an algorithm for smart home system to interact with the sensor as information source or trigger device and give the best service in a specific location whenever it is needed. In the smart home system, there are specific rules stored in the master station which have to be executed to provide the right service. In the centralized architecture, the master station store all the rules and executed those rules centrally.

The typical simple centralized smart home system is shown in Figure 1. This type of smart home system can be mentioned as centralized smart home system (CSHS). In CSHS, the rules execution mechanism is performed by the master station. The information or trigger will be provided to the master station by the trigger device. The period of each trigger is different. When the master station obtains the trigger from trigger device, it will provide the service to the home resident based on the algorithm it used. The service is a command from master station to turn on or turn off a device or set the device in specific state.

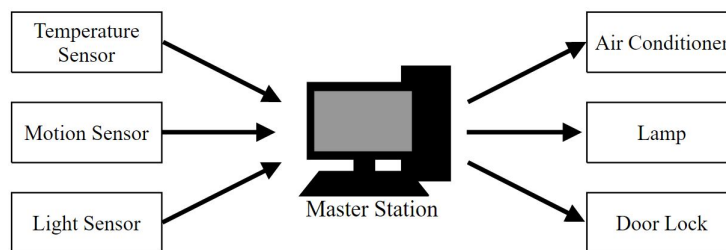


Figure 1. Centralized smart home system

The centralized mechanism is easy to manage and to configure. Because the user-system interaction can be performed through only one place/device to manage all devices in smart home system. However, in fault-tolerant system paradigm it produces a problem. The entire system will fail, if the master station fails. Another problem of CSHS is centralized mechanism provides many data-flows. This condition makes the system has a greater delay time. To solve the problem, we propose a distributed rule execution mechanism (DREM). Compared to the centralized mechanism, the DREM allows a device to provide its service without any commands from the master station. A service device can obtain a command from any trigger devices to provide service. In this mechanism, since the information is transmitted directly from trigger device to the service device, the data-flow and delay-time can be decreased.

2. CENTRALIZED SMART HOME SYSTEM

Many researchers have been performed studies on the CSHS [17]-[21]. The most recent and popular platform for this architecture is trigger-action platforms. These platforms are web-based systems that connect together some online services to give the users the capability to create and manage the rules [22], e.g., If-This-Then-That, Microsoft Flow, and Zapier. Since these platforms connect the actual physical devices to the online services, the security aspect is a must. To prevent the attackers to manipulate users data and devices, a study entitled decentralized action integrity for trigger-action IoT platforms proposed a security mechanism by using transfer tokens (XTokens) [23]. This token is a small trusted client installed on the user's device uses an XToken to automatically obtain a rule-specific token, which it transmits to the cloud service for rule execution. The decentralized term in this research refers the security mechanism of the system, not the rule execution mechanism. Therefore, we categorize this architecture as a CSHS.

The model of CSHS consists of three main parts. The first part is master station. Master station is a center of CSHS that collects the information, stores the rules, applies the algorithm, and controls every single devices inside and outside of the house. The second part is trigger. Trigger is the set of condition that makes the CSHS provides a service. Trigger is a general term that we use to determine all the condition that provide data to the master station (e.g., temperature sensor acquire the temperature in the living room, light sensor detect the low light ambient in the workspace). Trigger is a set of condition of trigger device. It can be a single

device or many devices, and the last part is service. Service is a set of condition in the smart home environment which is provided by master station and service device (e.g., turning the lamp on, set the temperature to 20 °C). Figure 2 shows the schema of this model.

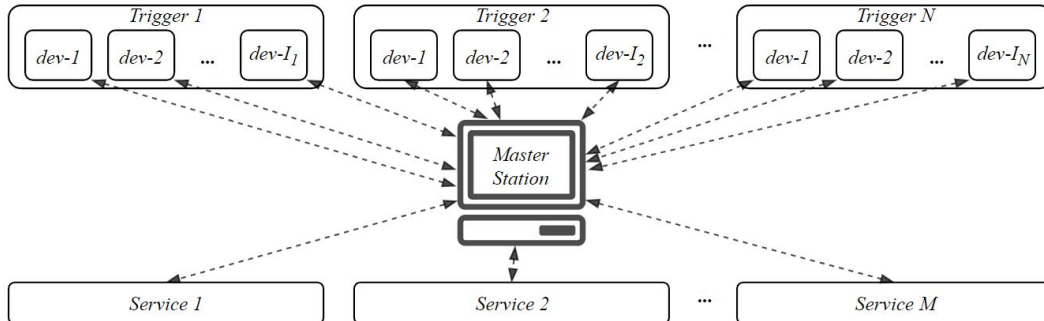


Figure 2. The model of centralized smart home system

From the Figure 2 we can see the system consists of many triggers and many services. And the function of the master station is to provide the service based on the trigger obtained from the trigger. The rule to provide the service in smart home can be written as an simple IF-THEN rule between trigger and service. Let M be the set of service and N be the set of triggers, the rule to provide the service in the smart home system can be written:

$$\forall m \in M, \forall n \in N : \forall m \exists n Q(m, n) \tag{1}$$

$$Q(m, n) = (n \Rightarrow m) \wedge (\neg n \Rightarrow \neg m) \tag{2}$$

since the n consists of the trigger devices I_n , where I_n can be single or multiple devices, the value of n is depend on the value of trigger devices I_n . Therefore, we can write in mathematical notation:

$$\forall n \exists I_n Q(n, I_n) \tag{3}$$

$$Q(n, I_n) = (I_n \Rightarrow n) \wedge (\neg I_n \Rightarrow \neg n) \tag{4}$$

The value of trigger device I_n is TRUE if the value of all elements in trigger n are TRUE. What it means by TRUE is that the real value of each trigger device matches to the stored value in the rule database. Furthermore, from (2) and (4) we can inference that:

$$(I_n \Rightarrow m) \wedge (\neg I_n \Rightarrow \neg m) \tag{5}$$

However, in the CSHS, the rule that is represented by (5) can not be performed directly because there is not direct path from I_n to m . Therefore, to provide a service in CSHS we have to use the (2) and (4). The algorithm to provide the service based on these two equations is shown in algorithm 1.

The algorithm 1 shows that in service provision, master station need to find the value of the trigger based on the data from the device and then find the service based on the trigger. This mechanism will produce delay time in service provision. The delay time of service provision can be decomposed as send time (S), access time (S), transmission time (T), propagation time (P), reception time (RP), receive time (RC) [24], and processing time (PC). To simplify the analysis, we can write send time, access time, and transmission time as a variable of X . And we can write reception time and receive time as a variable of Y .

$$X = S + A + T \tag{6}$$

$$Y = RP + RC \tag{7}$$

To obtain general equation of the delay time of service provision in Figure 3, we need to analyze four cases of the service provision: i) the provision of single service which is triggered by a single device,

ii) the provision of single service which is triggered by many device simultaneously, iii) the provision of many services simultaneously which each is triggered by a single device, and iv) the provision of many services simultaneously which each is triggered by many devices simultaneously. Based on the analysis of the four cases, we can write the delay time of service provision in general term:

$$D_{|M|,|N|} = X_{1,1} + P_{1,1,0} + Y_0 \sum_{n=1}^{|N|} I_n + \sum_{n=1}^{|N|} \sum_{i=1}^{I_n} PC_{0,n,i} + |M|X_0 + P_{o,|M|} + Y_{|M|} \tag{8}$$

where $|M|$ is the cardinality of set M , $|N|$ is the cardinality of set N , $D_{|M|,|N|}$ is the delay time needed to provide $|M|$ services which is triggered by $|N|$ triggers, $X_{n,i}$ is the delay time needed by the device i in trigger n to send the data, $P_{n,i,0}$ is a propagation delay from device i in trigger n to the master station (which is denote as 0), Y_0 is reception and receive time in master station, I_n is the number of device in trigger n , $PC_{0,n,i}$ is the processing time in master station (which is denote as 0) for the data obtained from device i in trigger n , X_0 is the delay time needed by master station to send the data, $P_{o,|M|}$ is propagation time from master station to service device $|M|$, and $Y_{|M|}$ is reception and receive time in service device $|M|$. From (8) we can know that for a single service the delay time is not that significant. But if the master station perform many services simultaneously, the delay time will be significant enough to affect the quality of services.

Algorithm 1 Service provision in centralize smart home system

```

1:  $data_{n,i} \leftarrow$  receive data from trigger  $n$ , device  $i$ 
2:  $n \leftarrow$  trigger  $n$ 
3:  $I_n \leftarrow$  find all devices in trigger  $n$ 
4: for  $i = 1$  to  $I_n$  do
5:   if ( $data_{n,i} == rule_{n,i}$ ) then
6:      $n\_value \leftarrow$  TRUE
7:   else
8:      $n\_value \leftarrow$  FALSE
9:   end if
10: end for
11:  $m \leftarrow$  Find service index for trigger  $n$  with  $n\_value$ 
12:  $m\_value \leftarrow$  service value
13: Provide service  $m$  with  $m$  value
    
```

▷ Variable n is the address or index of the trigger

▷ Variable n_value is the value of trigger n

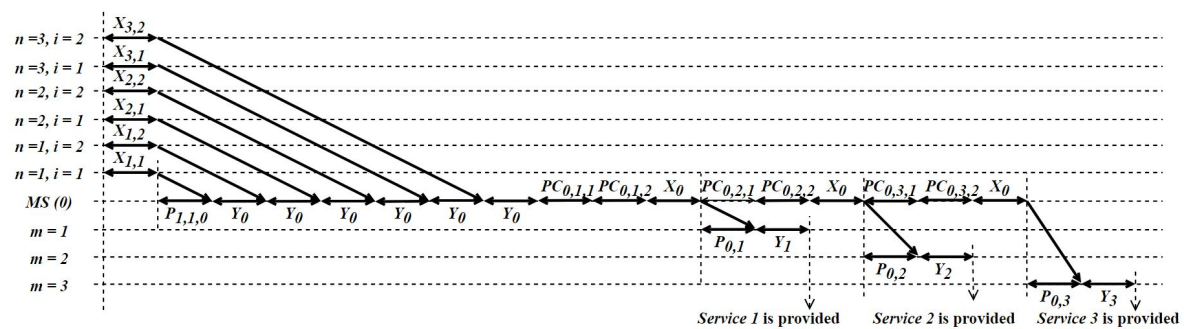


Figure 3. The delay time of service provision in CSMS

3. PROPOSED MECHANISM

To overcome the fault tolerance and delay time problems, we propose the DREM. This idea is first present in [25]. However this research is based on practical analysis without a proper mathematical rigor to model the mechanism. In this research, we proposed a more proper model and analysis related to DREM. To realize the DREM we need to add the direct path from trigger device to service. Figure 4 shows the addition of the path to the CSMS. The work principle of the model can be explained using the following illustration case. At 5 p.m. the home resident come home from work. At the gate, a sensor (trigger device) detects the home resident and send the data to master station and then the master station find the correct rule for this context

and provide the service to open the gate (service device). In the DREM, the sensor can directly transmits the data to the gate to perform a service to open the gate. How does the sensor know that it has to send the data to the gate? In the configuration time, master station need to broadcast the rule to all devices in the smart home system based on the rules they are involved in so that every single device knows what to do when they get a new data.

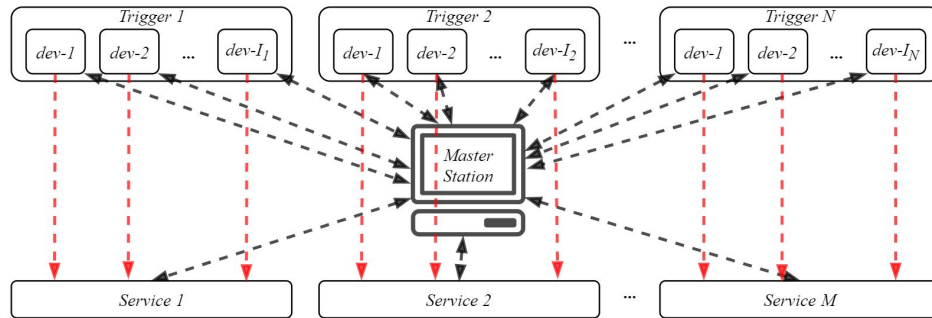


Figure 4. DREM

Another illustration is when the home resident enter the home and temperature is high. The motion sensor will detects the human presence while the temperature sensor detects that the temperature is high. Based on the rule, this two conditions are in the similar trigger and have to provide a service to turn on the air conditioner. In the DREM, both motion sensor and temperature sensor already knew that the service they have to trigger is turning on the air conditioner. Therefore, both of these sensor directly send data to the air conditioner and then the air conditioner checks the rule. If both condition is satisfied then the service is provided. However, if only one rule is satisfied (e.g, only temperature sensor send the temperature data, and motion sensor does not detects the human presence) then the service is not provided.

Based on these illustration, the system need some algorithms to: i) distributing the rules from master station to the trigger device and service device, ii) storing the rules from master station in trigger device and service device, iii) sending the trigger to right service device, and iv) receiving the trigger and performing the service. In this DREM, we assume that the behavior of the devices and the way to transmit and receive data is not different from CSHS. Therefore, there is no addition in energy consumption. Furthermore, the proposed mechanism focus on the algorithm in the application layer, so that all devices with the various communication hardware can adopt this algorithm. The first algorithm to distribute the rules to trigger device and service device is shown in algorithm 2. From the algorithm, we can see that each device will only get the rule for the service they are involved in. Therefore, the processing time in each device will not cause a larger delay time.

Algorithm 2 Distributing the rules from master station to the trigger device and service device

```

1: for  $n = 1$  to  $|N|$  do
2:    $I_n \leftarrow$  find all devices in trigger  $n$ 
3:    $m \leftarrow$  service  $m$  address
4:    $m\_value \leftarrow$  service  $m$  value
5:   for  $i = 1$  to  $I_n$  do
6:      $dev_i \leftarrow$  address of device  $i$ 
7:      $val_i \leftarrow$  value of device  $i$ 
8:     send  $val_i, m,$  and  $n$  to  $dev_i$ 
9:     send  $dev_i, val_i, m\_value,$  and  $n$  to  $m$ 
10:  end for
11: end for

```

▷ Variable m is the address or index of the service

Algorithm 3 shows the way to store the rules from master station into trigger device and service device. The trigger device will store the the value of the device, service address, and trigger index in the rule storage. While the service device will store device address, device value, service value, and trigger index. The storage in the trigger or service device is usually a build in linear storage (e.g, electrically erasable programmable read-only memory (EEPROM)). Algorithm 4 shows the way to trigger the service from trigger device. After

the device complete the task to acquire the data then the device will check if the data is similar or not to the previous data. If the data is different then it will send the data to service device based on the rule which is stored.

Algorithm 3 Storing the rules from master station in trigger device and service device

```

1: device receives packet from master station
2: if (device address == packet destination address) then
3:   read the packet
4:   if (device == trigger device) then
5:     store the val, m, and n in the device storage
6:   else
7:     store the dev, val, m-value, and n in the device storage
8:   end if
9: end if

```

Algorithm 4 Sending the trigger to right service device

```

1: val ← measurement result
2: if (val ≠ previous val) then
3:   find the val in the stored rule
4:   if (val exists in stored rule) then
5:     find the m and n based on the val address
6:     send the val and n to m
7:   end if
8: end if

```

The way to performing a service in service device is shown in algorithm 5. After receiving the trigger packet from trigger device, service device will store the data to the data storage. And then it will compare the number of device of the trigger in the data storage and rule storage. If the number of the trigger, both in data and rule are similar then the service device will check the similarity between data for all device and the rule. If all of them are similar then the service will be performed. However, if only one is similar and the others are different then the service will not be performed. The application layer of algorithms 2-5 is depend on the hardware used as trigger device and service device.

Algorithm 5 Receiving the trigger and performing the service

```

1: device receives packet from trigger device
2: store dev, val, and n in data storage
3: a ← count the number of device in trigger n in the data storage
4: b ← count the number of device in trigger n in the rule storage
5: if (a == b) then
6:   for i = 1 to a do
7:     if (datai == rulei) then
8:       service ← TRUE
9:     else
10:      service ← FALSE
11:    end if
12:  end for
13:  if (service == TRUE) then
14:    perform the service
15:  end if
16: end if

```

From the Figure 4 and algorithms 2-5 we can obtain that the service provision in the DREM works in parallel. Because of this mechanism, the delay time of the service provision in the DREM is also smaller than the one in CSHS. Figure 5 shows the delay time schema of the DREM. Based on this figure delay time to perform set of service which are triggered by set of trigger in the DREM can be written as (9).

$$D_{|M|,|N|} = \max \left[X_{n,1} + P_{n,1,m} + Y_m I_n + \sum_{i=1}^{I_n} PC_{m,n,i} : n \in N \wedge m \in M \right] \quad (9)$$

In (9) shows that the delay time to provide set of service which are triggered by set of trigger in the DREM are smaller than the one in CSHS. In this equation, the part that involve the master station is reduced. So that the propagation time from trigger device to the master station and from master station to the service device do not exist anymore. As a replacement, there is propagation time between trigger device and service device. The receiving time, processing time, and sending time in master station is also gone. As a replacement, there is a processing time in service device. And also since the DREM works in parallel, we only aware of the maximum delay of a single service provision.

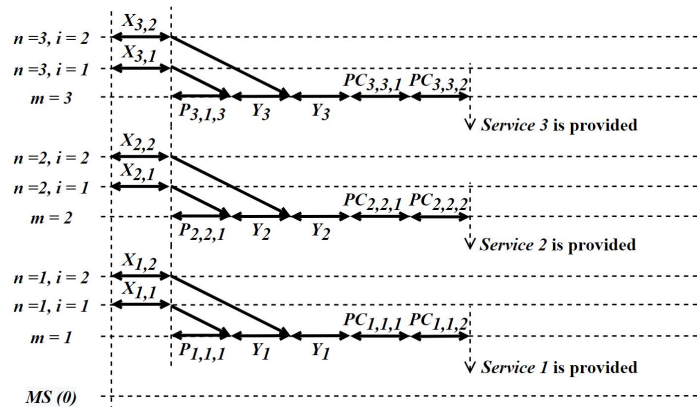


Figure 5. The delay time analysis of DREM

4. EXPERIMENT RESULT

To demonstrate the functionality of the DREM, the system is implemented using real hardware. The DREM implementation is based on the the block diagram shown in Figure 4. All devices (trigger device and service device) are represented by a TTGO LoRa32 V2 module. This module is equipped with ESP32 as a microcontroller, wireless communication module (LoRa, Bluetooth low energy (BLE), and WiFi), and LCD OLED display. This hardware module can be used to model both the trigger device that sends the data and the service device that receives the data and perform a service.

In this experiment, there are three scenarios that will be tested: i) single service triggered by single trigger device, ii) single service triggered by two trigger device, and iii) The first and second scenario are performed simultaneously. To perform this experiment, the first step is to write the rule for these two services in master station. In master station, the rule is stored in two tables (service rule and trigger rule) in a database. The rule of of this experiment is shown in Tables 1 and 2. Table 1 shows the service rule. In this table there are four variables: trigger index (n), trigger value (n_value), service index (m), and service value (m_value). As explained in the (2), the rule in Table 1 *IF-THEN* condition to provide the service (e.g., the rule in first row means IF trigger 1 is 1 THEN service 1 is 1. Table 2 shows the device its value in each trigger (e.g., the value of trigger 2: IF device 2 is 1 and device 3 is 30 THEN trigger 2 is 1.

Table 1. Service rule

n	n_value	m	m_value
1	1	1	1
2	1	2	1

Table 2. Trigger rule

i	i_value	n
1	1	1
2	1	2
3	30	2

The first experiment is to distribute the rules to the trigger device and service device. This experiment use the algorithm 2 in the master station side and algorithm 3 in device side. In the device the rule will be stored

in the EEPROM. Table 3 shows the result of this experiment. In Table 3, we can see that the trigger device and service device stored the different data. The trigger device stored the device value (i_value), service index (m), and trigger index (n) in consecutive address. The end of file of the rule will be marked with the value of 255. While in the service device, the data that will be stored in consecutive address are device (i), device value (i_value), service value (m_value), and service index (n). We can see that service device 1 stores only one entry of rule because this service is triggered by only one device. Whereas the service device 2 stores two entries of rule because this service is triggered by two devices. From this result we can obtain that the master station can distribute the rule successfully to the trigger device and service device.

Table 3. The rules that stored in the EEPROM

Address	$i = 1$	$i = 2$	$i = 3$	$m = 1$	$m = 2$
0	1	1	30	1	2
1	1	2	2	1	1
2	1	2	2	1	1
3	255	255	255	1	2
4	0	0	0	255	3
5	0	0	0	0	30
6	0	0	0	0	1
7	0	0	0	0	2
8	0	0	0	0	255

The next experiment is the DREM. There are two steps in this experiment. First, we run the system CSHS mode. The trigger device transmits the data to master station and then the master station executes the rule and provide the service. Second, we run the system in DREM mode. The correlated trigger device directly trigger all the services. We turn off the master station in this step. Table 4 shows the result of the experiment. From this result, we can know that the DREM works well. The system still can provide the right services even though we turn off the master station.

Table 4. The comparison of functionality performance of CSHS and DREM

Activity	CSHS	DREM
trigger device 1 is active	service 1 is active	service 1 is active
trigger device 2 is active and trigger device 3 is inactive	service 2 is inactive	service 2 is inactive
trigger device 2 is active and trigger device 3 is active	service 2 is active	service 2 is active

5. CONCLUSION

The proposed DREM can overcome the fault-tolerant problem in CSHS. The DREM can become an additional mechanism to the CSHS. This new mechanism can mask the failure of the system caused by the failure of the master station. The experiment results show that the DREM can mask the failure in the master station by directly transmit the data from trigger device to service device. This mechanism makes the services provision without master station possible. The mathematical analysis also shows that the service provision delay time of the DREM is less than the delay time of CSHS. Our future work is to synchronize the time in each device so that the delay time can be measure in the real system.




REFERENCES

- [1] J. Waleed, A. M. Abduldaim, T. M. Hasan, and Q. S. Mohaisin, "Smart home as a new trend, a simplicity LED to revolution," *2018 1st International Scientific Conference of Engineering Sciences - 3rd Scientific Conference of Engineering Science (ISCES)*, 2018, pp. 30-33, doi: 10.1109/ISCES.2018.8340523
- [2] Y. Kashimoto, M. Fujiwara, M. Fujimoto, H. Suwa, Y. Arakawa, and K. Yasumoto, "ALPAS: analog-PIR-sensor-based activity recognition system in smarhome," *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 880-885, doi: 10.1109/AINA.2017.33.
- [3] W. Yang, X. Jing, and H. Huang, "An intelligent human behavior-based reasoning model for service prediction in smart home," in *IEEE Access*, vol. 6, pp. 68535-68544, 2018, doi: 10.1109/ACCESS.2018.2879904.
- [4] V. Milykh, D. Vavilov, I. Platonov, and A. Anisimov, "User behavior prediction in the "offline" smart home solutions," *2016 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, 2016, pp. 24-27, doi: 10.1109/ZINC.2016.7513646.




- [5] P. Mtshali and F. Khubisa, "A smart home appliance control system for physically disabled people," *2019 Conference on Information Communications Technology and Society (ICTAS)*, 2019, pp. 1-5, doi: 10.1109/ICTAS.2019.8703637.
- [6] K. Paridari, A. Parisio, H. Sandberg, and K. H. Johansson, "Robust scheduling of smart appliances in active apartments with user behavior uncertainty," in *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 247-259, Jan. 2016, doi: 10.1109/TASE.2015.2497300.
- [7] G. Stratogiannis *et al.*, "User and home appliances pervasive interaction in a sensor driven smart home environment: the SandS approach," *2015 10th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, 2015, pp. 1-6, doi: 10.1109/SMAP.2015.7370088.
- [8] P. K. Madupu and B. Karthikeyan, "Automatic service request system for security in smart home using IoT," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1413-1418, doi: 10.1109/ICECA.2018.8474684.
- [9] S. U. Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/internet-of-things," *2018 Fifth International Conference on Software Defined Systems (SDS)*, 2018, pp. 126-129, doi: 10.1109/SDS.2018.8370433.
- [10] H. Singh, V. Pallagani, V. Khandelwal, and U. Venkanna, "IoT based smart home automation system using sensor node," *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, 2018, pp. 1-5, doi: 10.1109/RAIT.2018.8389037.
- [11] B. R. Stojkoska and K. Trivodaliev, "Enabling internet of things for smart homes through fog computing," *2017 25th Telecommunication Forum (TELFOR)*, 2017, pp. 1-4, doi: 10.1109/TELFOR.2017.8249316
- [12] M. Uddin and T. Nadeem, "EnergySniffer: home energy monitoring system using smart phones," *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012, pp. 159-164, doi: 10.1109/IWCMC.2012.6314195
- [13] L. Liu, Y. Liu, L. Wang, A. Zomaya, and S. Hu, "Economical and balanced energy usage in the smart home infrastructure: a tutorial and new results," in *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 556-570, Dec. 2015, doi: 10.1109/TETC.2015.2484839
- [14] B. Rajasekhar and N. M. Pindoriya, "Multi-stage scheduling for a smart home with solar PV and battery energy storage — a case study," *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, 2015, pp. 1-6, doi: 10.1109/ISGT-Asia.2015.7386984.
- [15] Z. Hong, P. Li, and W. Jingxiao, "Context-aware scheduling algorithm in smart home system," in *China Communications*, vol. 10, no. 11, pp. 155-164, Nov. 2013, doi: 10.1109/CC.2013.6674219.
- [16] W. Jingxiao, Z. Hong, and S. Nigang, "Research on context-aware scheduling algorithm based on correlation in smart home environment," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops Ph.D. Forum*, 2012, pp. 2312-2315, doi: 10.1109/IPDPSW.2012.289.
- [17] A. Banerjee, F. Sufyanf, M. S. Nayel, and S. Sagar, "Centralized framework for controlling heterogeneous appliances in a smart home environment," *2018 International Conference on Information and Computer Technologies (ICICT)*, 2018, pp. 78-82, doi: 10.1109/INFOCT.2018.8356844.
- [18] F. D. Jivani, M. Malvankar, and R. Shankarmani, "A voice controlled smart home solution with a centralized management framework implemented using AI and NLP," *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1-5, doi: 10.1109/ICCTCT.2018.8550972.
- [19] S. P. Lim and G. H. Yeap, "Centralised smart home control system via XBee transceivers," in *2011 IEEE Colloquium on Humanities, Science and Engineering*, 2011, pp. 327-330, doi: 10.1109/CHUSER.2011.6163743.
- [20] M. B. Krishna and A. Verma, "A framework of smart homes connected devices using internet of things," *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016, pp. 810-815, doi: 10.1109/IC3I.2016.7918794.
- [21] S. M. Brundha, P. Lakshmi, and S. Santhanalakshmi, "Home automation in client-server approach with user notification along with efficient security alerting system," *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2017, pp. 596-601, doi: 10.1109/SmartTechCon.2017.8358441.
- [22] B. Ury *et al.*, "Trigger-action programming in the wild: an analysis of 200,000 IFTTT recipes," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3227-3231, doi: 10.1145/2858036.2858556.
- [23] E. Fernandes, A. Rahmati, J. Jung, and A. Prakash, "Decentralized action integrity for trigger-action IoT platforms," in *Proceedings 2018 Network and Distributed System Security Symposium*, Internet Society, 2018, pp. 1-16, doi: 10.14722/ndss.2018.23119.
- [24] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 147-163, 2002.
- [25] A. S. Budi, H. Fitriyah, E. Setiawan, R. Primananda, and R. Maulana, "Distributed rule execution for smart home system," in *Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology*, 2020, pp. 183-189, doi: 10.1145/3427423.3427458.

BIOGRAPHIES OF AUTHORS






Agung Setia Budi    spent his first higher education at the electrical engineering department, the University of Brawijaya from 2005 to 2010. He obtained a master's degree in Electrical and Electronic Engineering separately from the University of Brawijaya and the University of Miyazaki in 2012 and 2013, respectively. He continued his doctoral study in Interdisciplinary Graduate School of Agriculture and Engineering at the University of Miyazaki in 2015, and got the degree in 2019. Now He is an active academic staff in the Faculty of Computer Science, Universitas Brawijaya, Indonesia. His current research interest includes analog and digital circuit for biological signal processing, embedded system, smart devices, smart home system, internet of things, and wireless sensor network. He can be contacted at email: agungsetiabudi@ub.ac.id.






Hurriyatul Fitriyah    is Assistant Professor in Computer Engineering, Informatics Engineering Department, Faculty of Computer Science, Universitas Brawijaya, Malang Indonesia. She received her Bachelor Degree in 2007 from Physics Engineering Department, Institut Teknologi Sepuluh Nopember, Surabaya Indonesia. She received her Master Degree in 2012 from Electrical and Electronics Engineering Department, Universiti Teknologi Petronas, Perak Malaysia. Her research interest is in the Computer Engineering, Computer Vision and Machine Learning. She has several Journals and Conferences Publication, Books and International Patent. She can be contacted at email: hfitriyah@ub.ac.id.






Eko Setiawan    received Bachelor Degree from the Department of Electrical Engineering – Universitas Brawijaya, Malang, East Java, Indonesia in 2009. He obtained a master's degree in Electrical and Electronic Engineering separately from the University of Brawijaya and the University of Miyazaki in 2012 and 2013, respectively. He continued his doctoral study in Interdisciplinary Graduate School of Agriculture and Engineering at the University of Miyazaki in 2015, and got the degree in 2018. Now He is an active academic staff in the Faculty of Computer Science, Universitas Brawijaya, Indonesia. His current research interest includes Embedded Systems, Control Systems, Robotics, Intelligent Device Engineering. He can be contacted at email: ekosetiawan@ub.ac.id.



Rakhmadhany Primananda    received Bachelor Degree from the Department of Electrical Engineering – Universitas Brawijaya, Malang, East Java, Indonesia in 2010. Then, he was completed Master Degree from the Department of Informatics - Sepuluh Nopember Institute of Technology, Surabaya, East Java, Indonesia in 2014. Currently working as a lecturer at Computer Engineering Study Program and as a member at Information-Centric Networking research group - Department of Informatics, Faculty of Computer Science – Universitas Brawijaya. His research interest is Wireless Sensor Network, Internet of Things, and Fault-Tolerant Computer System. He can be contacted at email: rakhmadhany@ub.ac.id.



Rizal Maulana    is a lecturer at the Computer Engineering, Faculty of Computer Science, Brawijaya University. He is also a coordinator of Smart Systems Research Group. He received his Bachelor Degree in Electrical Engineering from Brawijaya University Indonesia, in 2011. His Master Degree in Electrical Engineering received from National Central University Taiwan and Brawijaya University Indonesia (Joint Degree Program) in 2014. His researches are in the fields of biomedical engineering and robotics. He can be contacted at email: rizal_lana@ub.ac.id.