

Web Server-based Distributed Machine Socialization System

Changsu Kim¹, Hankil Kim², Jongwon Lee³, Hoekyung Jung⁴

^{1,3,4}Department of Computer Engineering, PaiChai University, Doma2-Dong, SeoGu, DaeJeon, Korea

²Korea University of Media Arts, 312, Daehak-gil, Sejong, Korea

Article Info

Article history:

Received Dec 9, 2017

Revised Mar 12, 2018

Accepted Mar 30, 2018

Keyword:

IoT

Machine socialization

Micro controller

Schema

Sensor

ABSTRACT

In recent years, there has been an increasing trend of offering services that are useful to users, such as Google's Nest, through machine socialization between parts and devices in specific spaces such as automobiles, homes, and factories. The existing inter - device collaboration system is a centralized system using router, and it controls collaboration between devices by building OpenWrt and web server on router. However, due to the limited hardware resources on the router, it generates network traffic congestion as the number of requests from the client increases or the number of clients connected to the server increases. In this paper, we propose a distributed machine collaboration system based on web server using inter - device collaboration algorithm. The study of Micro Controller Unit (MCU) has reduced the traffic incidence by solving the request sent to the router from each device by oneself.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Hoekyung Jung,
Departement of Computer Engineering,
Paichai University,
155-40 University Road, Seogu, DaeJeon, Korea.
Email: hkjung@pcu.ac.kr

1. INTRODUCTION

Recently, interest in IoT has been growing. For example, Google's home temperature control system 'Nest', and 'On Hub', which provides a wireless LAN service by selecting an efficient channel with 13 antennas [1-3]. Research on platform, network, security, and smart home on which IoT is based are being conducted. The existing collaborative system is a centralized system and OpenWrt is built on the router and web server such as APMsetup or IIS is constructed and utilized [4-7]. However, if there are more clients connecting to the router server or the number of requests from clients increases, data loss and bottlenecks may occur, resulting in longer response time or interruption of operations. In this paper, we propose a web server based distributed machine socialization system which will overcome the problem of insufficient resources of the router by attaching MCU to each machine. Also, to minimize user interference, Json schema is received from web server and information of machine is input automatically. This enables smooth operation between machine unlike the existing system, and even if a new machine is added, the changes can be perform without interfering with the operating environment.

2. DISTRIBUTED MACHINE SOCIALIZATION SYSTEM DESIGN

The proposed system is based on Wi-Fi. Sensor layer that receives sensor data by using Arduino and stores it in the database. Application layer which stores Relational Data generated from User Application in database. Raspberry Pi is composed of Machine Control Layer. Figure 1 shows the system architecture including the modules used in the proposed system.

The sensor data generated by the sensor is transferred to Arduino and is always stored in the database. When a new machine required for the operation is connected to the same WiFi network connect module, the Json schema structure required for collaboration between the machines is loaded through the Web server loaded on the Master Raspberry Pi. The schema has a structure that can store the information of the machine, and the data to be put into the structure is inputted through the information of the machine. Once the data is loaded into the schema structure, it starts to work.

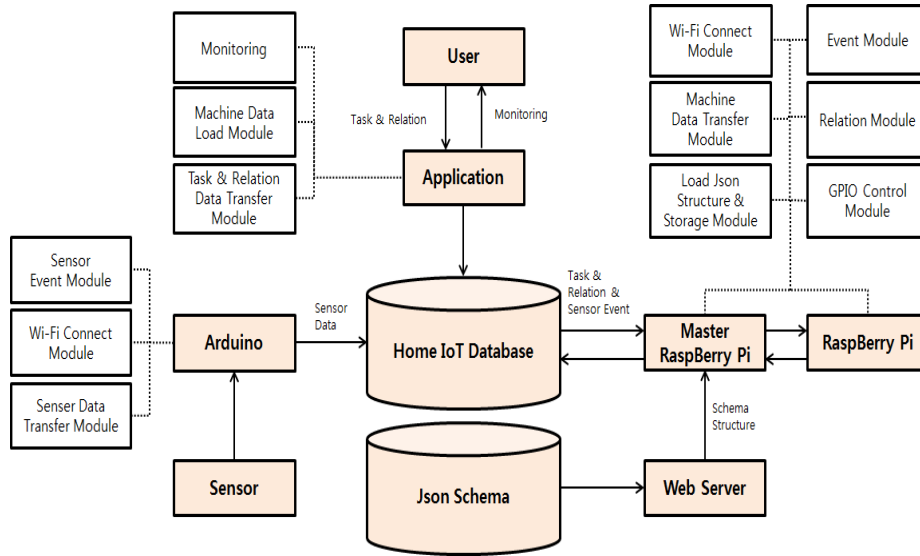


Figure 1. Machine Socialization System Architecture

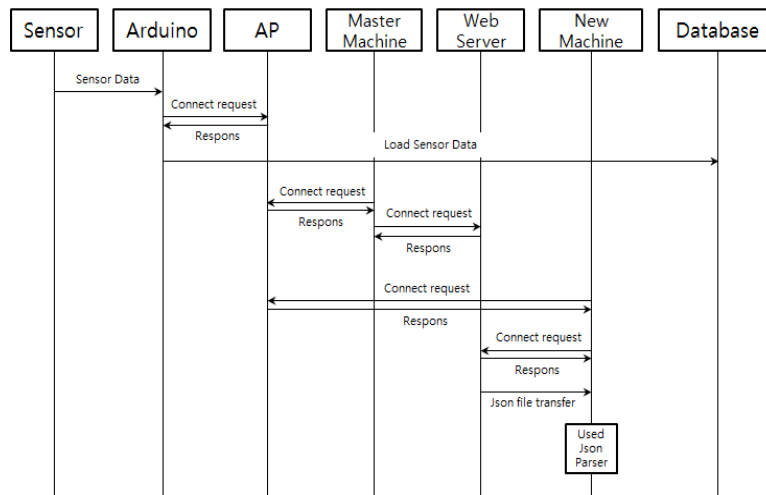


Figure 2. Add New Machine in to Machine Socialization System of Sequence Diagram

Figure 2 is a sequence diagram showing the process of adding a new machine required for the task. When the Arduino connected to the sensor is connected to the AP, Arduino always loads the data from the sensor into the database. Once the Master Machine (Raspberry Pi) is connected to the AP, it connects to the Web server. The web server includes the Json schema. When a new machine is added, the web server automatically transmits the Json schema structure to the machine, and the machine receives and stores the Json schema structure. As a result, the new machine uses the parser to load the information of the machine that it has on the Json schema.

When inputting task and correlation required for collaborative work is completed, it is stored in database. Figure 3 is a sequence diagram that receives the necessary data from the user for collaboration and finds the machine to work, and the next machine to work on, when the event occurs. The machine receives the Json schema structure from the web server, stores the machine information in ‘Machine.json’ in Storage, and sends the machine information to the ‘User Application’. The user can set the name of the task through the application and can be provided with the information of the machine that can be used for the task. You can select the desired machine from the machine list and set the order of the machine.

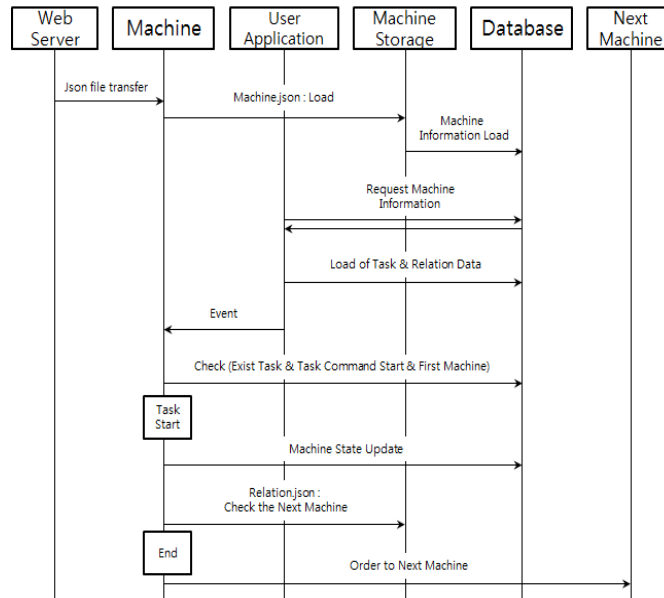


Figure 3. Machines Cooperation Sequence Diagram

When inputting task and correlation required for collaborative work is completed, it is stored in database. When an event occurs, Raspberry Pi checks for the, existence of a task, the status of the task command, and checking the first machine.

When each machine is connected to one network through a router, a collaboration algorithm between the machines is required in order for the machine to smoothly operate and finish.

Figure 4 is a flow chart that enables the machine to work collaboratively when an event occurs. It uses the recursive function in the Machine Control Layer to accept what is stored in the database from the User Application as an event. Therefore, collaboration between active machine can be achieved.

The first machine checks in real time whether an event called ‘task generation’ has occurred. When a task is created from an application, it waits for the user to execute the created task. When a user launches a task, the machines compare the information about the machine that the user first operates with the machine in the database, to the machine that the machine has in the Json schema. The first machine will start working and machines that are not on their turn will wait until they receive the first machine’s shutdown signal. When the operation of the machine is finished, the information of the next machine is checked in the database, and if there are further machines to be executed in the task, a start message is transmitted to the next machine in the task list. To find the machine, we send the message using the IP address and TCP / IP communication stored in the table when updating the machine information to the database. The second machine that is waiting for the signal from the previous machine will start working when it receives a signal that the previous machine has stopped working.

The proposed inter-machine collaboration system is divided into Sensor Layer module and Control Layer, which are divided into User Application and Machine Control. Figure 5 shows the module structure of the inter-machine collaboration system. The sensor layer receives the data generated by the sensor through Arduino’s Sensor Event Module. When connected to Wifi via ‘Wifi Connect Module’, it is saved in database with ‘Sensor Data Transfer Module’. When a new machine is added in the Machine Control of the Control Layer, the Json Schema structure required for collaboration is fetched from the web server through the ‘Load Json Structure & Storage Module’ and stored in the storage. When the machine information enters the schema through the parser, the information of the machine is transmitted to the user through the Machine

Data Transfer Module. The user can check the information of the machine's stored through the 'Machine Data Load Module' as an application.

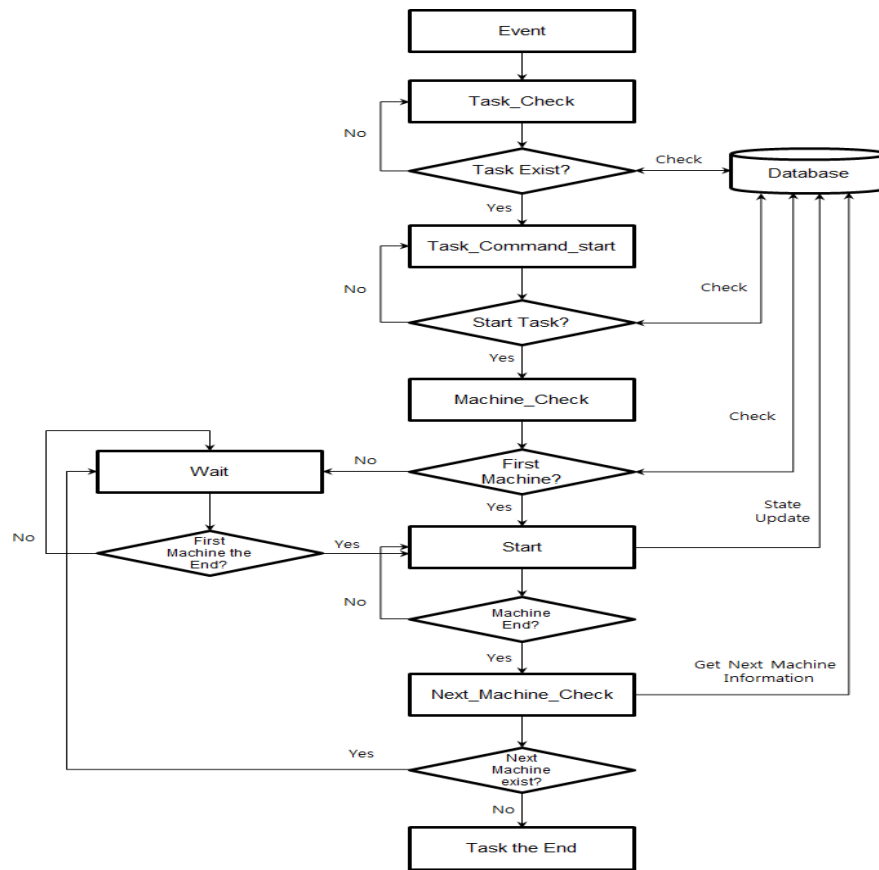


Figure 4. Machines socialization system algorithm diagram

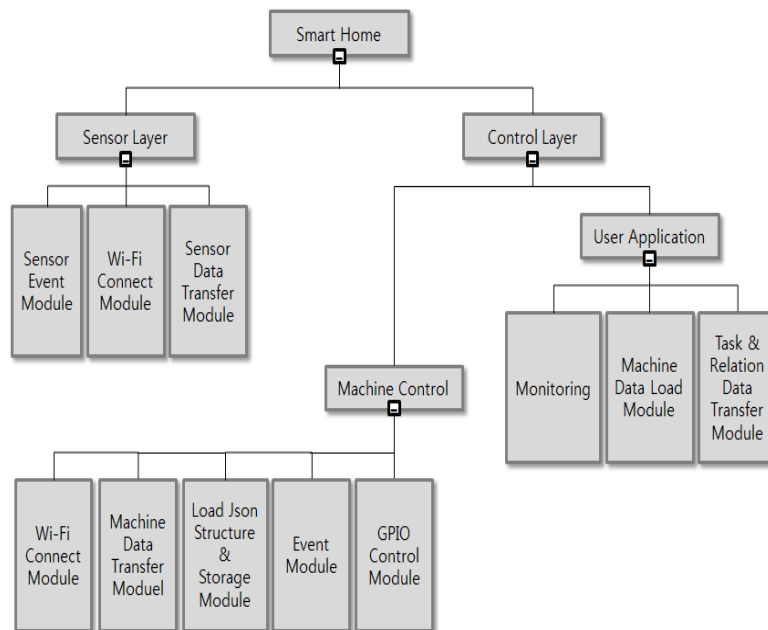


Figure 5. Structure of machine socialization system module

The application can define the job and select the equipment required for the operation through the information of the received machine. Once you have ordered the machines, you can transfer data via the ‘Task & Relation Data Transfer Module’. When an event occurs from the sensor, the machine operates through the ‘Event Module’ and the ‘GPIO Control Module’ controls the machine.

3. SYSTEM IMPLEMENTATION AND REVIEW

In this chapter, we describe the development environment and actual implementation required to implement the system designed in Chapter 2, and describe the proposed system. Table 1 shows the development environment for device control and sensor data collection in the proposed inter-device collaboration system.

Table 1. Example of mapping rule

Development Environment	
Board	Raspberry Pi 3 Model B, Arduino Uno
Operating System	Raspbian(Version : May 2016)
Database	MySQL
Access Point	iptime A2004NS-R
Development Tools	Arduino Sketch(Version : 1.6.9), Python 2(IDLE 2.7.9)
Development Language	C, PHP, Python, Json
Sensor	Temperature DS18B20
	Humidity CL-MM1001
	Infrared thermal MLX90614

The MCU for controlling the sensor layer is implemented using Arduino Uno, and the MCU for controlling the device in Machine Control Layer is implemented using Raspberry Pi 3. The database used to store the sensor data and device state was MySQL.



Figure 6. Products used in machines cooperation

Figure 6 shows the MCU that controls the devices and devices used to implement the scenario of the inter-device collaboration system, and the smartphone and sensor to activate the application.

The dotted line in Figure 7 shows the ‘response time’ between machine and the server in a Centralized system. We can see that with every addition of a new device in the centralized inter-device collaboration system, the response time keeps increasing. The solid line graph shows the result of the same system in the proposed distributed inter-device collaboration system under the same conditions.

Data requests were tested starting at 10 and increased to 50, and the response time was measured. In the case of the centralized system, the average response time according to the number of requested jobs was 0.6 seconds, and the bottleneck frequently occurred as the number of requested jobs increased. As we proceeded procedurally, it was confirmed that the response time became longer as more requests were made with every addition of a new system. On the other hand, in the proposed distributed machine collaboration system, the average response time decreased by 0.5 seconds from 0.6 to 0.1 second. This is because the server side device and the client side device are in 1: 1 communication and the other devices are waiting until they are connected to the server. As a result, the proposed distributed machine collaboration system mitigates existing data loss and bottlenecks and can facilitate collaboration among devices.

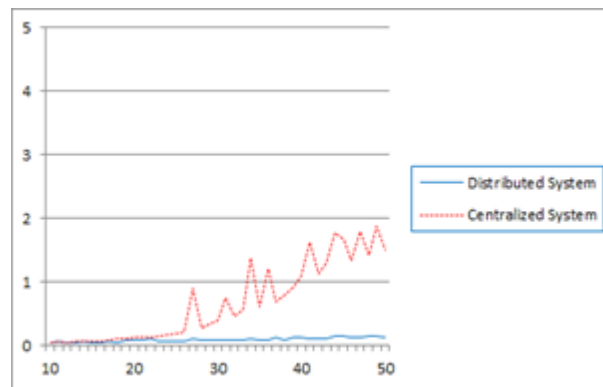


Figure 7. A Graph of the response time data

4. CONCLUSION AND FUTRUE RESEARCH

The existing collaboration system sets up OpenWrt on the router and installed APM (Apache PHP MySQL) on it to perform device control and device request operations. The control of all devices in the center has a merit of simple configuration. However, when a new device is added, the data is lost and the response time is lowered due to the low performance of the router and when the number of requests from the client is increased. In this paper, we implemented the device so that all devices inside the home can perform the role of Server and Client without using the resources of the router by attaching MCU to each device. By comparing and analyzing the existing centralized machine collaboration system and the distributed machine collaboration system, we found that the average response time was reduced from 0.6 seconds to 0.1 seconds a difference of 0.5 seconds. As a result, the problems of the system such as the response time and router resources between the devices are compensated.

Future research needs to be done to develop an appropriate specification platform, which can collaborate between machines, because the system implementation using MCU (Raspberry Pi 3) is over specifications.

ACKNOWLEDGEMENTS

This research was supported by The Leading Human Resource Training Program of Regional Neo industry through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(No. 2016H1D5A1911091).

This work was supported by the research grant of Pai Chai University in 2017.

REFERENCES

- [1] M. S. Gong, H. J. Che, B. H. Yu, "IoT technology trends and outlook," *The Korean Society of Mechanical Engineers*, vol. 56, no. 2, pp. 32-36, Feb. 2016.
- [2] S. H. Kim, "Internet of Things Technology," *The Institute of Electronics Engineers of Korea*, vol. 43, no. 3, pp. 64-71, Mar. 2016.
- [3] H. J. Lee, "Internet of Things Based Smart Home," *Korea Institute of Communication Sciences*, vol. 32, no. 4, pp. 44-49, Mar. 2015.

- [4] W. D. Jo, S. T. Choi, J. S. Baek, Y. G. Min, Y. G. Lee, G. P. Lee, G. C. Park, J. I. Lee, "Big data Sensing Technology based on Internet of Things," *Korea Institute of Communication Sciences*, vol. 32, no. 11, pp. 21-31, Oct. 2015.
- [5] J. Byun, S. Kim, Y. T. Shin, J. B. Kim, "Smart City Implementation Models Based on IoT Technology," *Advanced Science and Technology Letters*, vol. 129, no. 41, pp. 209-212, 2016.
- [6] Y. Fujiwara, K. Hashimoto, A. Rahim, P. Vlacheas, "Context Aware Services: A Novel Trend in IoT Based Research in Smart City Project," *IEEE*, pp. 479-480, Jul. 2015.
- [7] A. Singh, "A study of Architectural Elements of Internet of Things and its Applications, International Journal of Scientific Research in Science," *Engineering and Technology*, vol. 2, no. 2, pp. 1189-1194, Mar. 2016.
- [8] Syed Husain, Athul Prasad, Andreas Kunz, Apostolos Papageorgiou and J. S. Song, "Recent Trends in Standards Related to the Internet of Things and Machine-to-Machine Communications", *Journal of Information and Communication Convergence Engineering*, vol. 12, no. 4, pp. 228-236, 2014.

BIOGRAPHIES OF AUTHORS



Changsu Kim. He received his B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering of Paichai University, Korea, in 1996, 1998, and 2002, respectively. From 2005 to 2012, he worked for the Department of Internet at Chungwoon University as a professor. Since 2013, he has worked in the Department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, web 2.0, and the semantic web.



Hankil Kim. He received the B.S., M.S. degrees from the Department of Electronic Engineering of Hanbat National University, Korea, in 2006, and 2011, and the Ph. D. degree in 2015 from the Department of Computer Engineering of Paichai University, Korea. Since 2006, he worked for the Department of Music & Sound Technology at Korea University of Media Arts as a professor. His current research interests include information retrieval system, web 2.0, and the semantic web.



Jongwon Lee. He received the B.S., M.S. degree from the Department of Computer Engineering of Paichai University, Korea in 2014 and 2016. He is currently a Docorate course in Department of Computer Engineering of Paichai University. His current research interests include multimedia information processing, information retrieval system, and semantic web.



Hoekyung Jung. He received the M.S. degree in 1987 and Ph. D. degree in 1993 from the Department of Computer Engineering of Kwangwoon University, Korea. From 1994 to 1995, he worked for ETRI as a researcher. Since 1994, he has worked in the Department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, information processing, information retrieval, and databases.