

# Middleware for Multi-sensors Context-Aware System to Control the Smart Laboratory Room

Khamla Non-Alinsavath, Lukito Edi Nugroho, and Agus Bejo

**Abstract**—Context Awareness technology is one of the important technologies to implement the smart application and it increases productivity of smart home, health care, and mobile computing which context aware systems were successfully applied to those fields. Middleware for context aware system plays an important role in the distributed application and is required for context based through context aware technology and for accuracy context communication with various raw sensors. The complexity of the hardware system and different platforms, different systems are the main issues for current application. In this paper we introduce the development of Context-Aware Middleware as the purpose of providing an architecture and implement the middleware as being intermediary between the raw sensors and application layers. The middleware is presented to handle the complex system of the hardware layer, bridges between physical and application layers that we provide several kinds of sensors consisting of different systems to operate with prototype platforms together with modeling and managing the middleware in accordance with the system integrates sensing, computing and user interaction as the outcome of the monitoring and reporting user's physical performance by capture the movement in the sensor range.

**Index Terms**—Context awareness, middleware, control system, sensor network.

## I. INTRODUCTION

The advancement of smart application together with the sensor technologies has significantly developed to create context aware application, to recognize activities in the smart environment that bringing the encourage users to collect and share situational awareness along with human's behavior automatically and intelligently. Middleware for Context awareness is required for context based service through context aware technology and for accuracy context communication with various sensors such as motion sensor, temperature and humidity sensor, light sensors, can be aware of changes in the user's activities and surrounding by capture human movement in the sensor range [1].

Context awareness applications are increasingly used and developed due to support the various computing technologies as well as pervasive computing. The information is used to communicate the situation for an entities (people, places, and objects) that relevant between user and application and one is able to interpret context information, to interrupt explicit operation to the current situation which we called Context [2]. Context awareness is the ability of a program or computing

device to detect, sense, interpret, act and respond to an aspect of the environment such as location, time, temperature or user identity or from the perspective of adapting an application to context [3], [4]. According to the influence of the system behavior, context awareness categorization can be divided into active and passive context awareness:

- Active context awareness: an application automatically adapts to discover context through changing the application's behavior.
- Passive context awareness: an application presents the new or update context to an interested user or make the context persistent for the user to retrieve later [5].

In terms of Context awareness middleware has been done in trying to make application and architecture in pervasive computing context aware technology that's considered only for specific application at specific platform and they tried to design the middleware of context aware for personal healthcare information system. The Middleware had developed as intermediate level and being a centered of the system that make application communicate to the physical layer easily. The need for middleware assist data management, provide operating and set the system become one effectively application which could be centralized the multiple components [6]-[8]. In this paper is introduced the Context Awareness Middleware with the purpose to provide an architecture and develop the Middleware as being intermediary between the raw sensors (hardware layer) and application layer that we provide several kinds of sensors consisting of different systems operate with electronics prototype platform which we model and manage the middleware in accordance with the system integrates sensing, computing and user interaction as the outcome of monitoring and reporting user's physical performance by captured the moved in – out in the sensor range

Monitoring of human movement, behavior, routine has increasing y become significant issue. Most of the pattern algorithms which are available mainly used for capturing the moved in-out in the sensor range. In the article we focus on the subject of context aware middleware by using raw sensors such as Passive Infrared (PIR) sensors, Humidity and Temperature (DHT11) sensors and Light Dependent Resister (LDR) sensors to monitor and to implement the system that can be used for sensor network with the hardware technique has developed rapidly and complicated system. To improve the efficiency information checking and reduce the processing time to control system and to fulfill the requirements [9] of middleware such as heterogeneity, mobility, scalability, and ease of deployment and configuration. We propose to use context awareness technique to design and understand users for modeling human behavior, to achieve the high

Manuscript received February 20, 2016; revised August 10, 2016.

The authors are with Universitas Gadjah Mada, Yogyakarta, Indonesia (e-mail: khamla@mail.ugm.ac.id, lukito, agusbj}@ugm.ac.id).

performance capturing movement for sensor devices.

The rest of this paper is structured as follows: Section II introduces the related work; Section III describes the design and architecture of the middleware for Context awareness. Section IV describes the performance evaluation and the simulation and work result are mentioned in Section V, finally we draw the conclusion and future work for the paper in Section VI.

## II. RELATED WORKS

In the area of the Middleware Context-aware architectures that have been proposed already as the same layer conceptual framework protocols and service for sensing, raw input data, and management / processing of the contextual information. One of the most significant properties of context aware Middleware is to efficiently infer context from sensory data, acquisition, processing, and relevant context recognition and provide to application. Middleware lies between physical and application layers that support completely functions to the system. In this site we propose some comparison concerning the term of Middleware platforms for context aware computing, in which most of them are concerned with one or more aspects in an ad hoc manner. Context Toolkit [10] uses the approach of widget to get raw contextual information from sensors and passes it either to interpreters or to sever for aggression that use simple HTTP protocol. Communication and the XML as the language model for the context. MidCASE [10] was introduced with the purpose to give a service oriented middleware to bridge the gap between the programmable application layer consisting of different scenarios and the hardware layer consisting of extremely heterogeneity devices that Middleware utilizes a service-oriented, distributed extensible architecture to achieve the service in each awareness service domain, and the aware process is achieved by applying rule-based reasoning. MoCA [11] (Mobile Collaboration Architecture) has been used by several research groups in Brazil and abroad for the development of small scale or experimental context aware application, MoCA is a service based architecture which offers support for the development of distributed context aware application.

The Service-Oriented Context Aware Middleware (SOCAM) [12], Context-Awareness Sub-Structure (CASS) [13], and Context Managing Frameworks are centralized architectures that in general receive context data from distributed context sensor and offer it in mostly processed to the clients.

## III. DESIGN AND ARCHITECTURE OF THE MIDDLEWARE FOR CONTEXT AWARENESS

The middleware context-aware system was designed for supporting and providing the context to service in a smart laboratory with context sharing on wireless network to control and interface context request between different kind of sensors. With mediation architecture as a middleware which sense raw data from various sensors through prototype platforms.

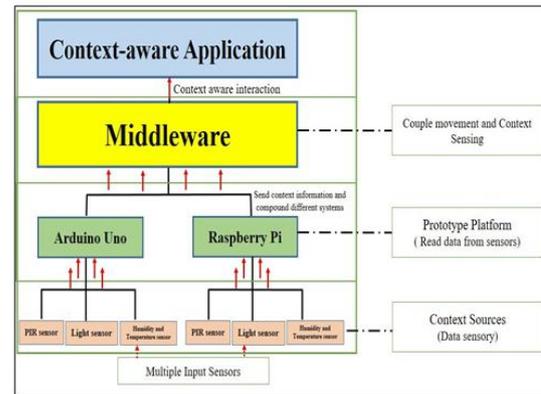


Fig. 1. The system design of middleware for multi-sensors context aware system.

Fig. 1 depicts the overall works of this research. The system is consisting of 4 components: Context sources layer is a physical layer that provide multiple input sensors to support the system and monitor the environment from light sensor, temperature and humidity sensor for controlling surrounding environment, the devices can detect when human has moved in or out of the sensors range in the detection area. Prototype platforms are responsible for reading data from sensors and control multiple sensors as two different systems, when collect information from context sources then prototype platform will send all information to the middleware.

## IV. CONTEXT AWARE MIDDLEWARE IMPLEMENTATION

### A. Context-Aware Middleware Components

To pass on the requirement of the system, we implement the automatic control context awareness to reduce the use of energy and when you turn on/turn off the lighting by this system, improve skills and experience to perform the system by using PIR motion sensor to capture the human movement behavior and pattern recognition monitoring. Light Sensor (LDR) and Temperature and Humidity sensors are used to control system about the lamp that can be sent alerts I the networks range. The system can capture human movement and behavior that we will put some activities such as lying, walking, working with electronic devices, along with this we put the temporal data like time and date in the system to set the time for controlling human behaviors.

Physical layer consists of multiple sensors namely Passive Infrared sensors, Light-Dependent Resistor, Temperature and Humidity sensors that use to capture and communicate with the environment. Two different types of platforms are used to read data from the above sensors and send information to the middleware layer to be ready for communicating to the application layer.

Middleware layer is consisted by three components such as Sensor Interface, Processing and Application service interface. Sensor interface is to communicate with hardware through protocols called TCP/IP (Transmission Control Protocols / Internet Protocols) and Serial Protocols to integrate, decode, sort and queue the data from physical layer. Processing would be received data from the hardware or received from sensor interface then the middleware will manage, control the data and collect data from the physical

layer. Finally the information is will be collected to application for providing context aware system and event processing service and Application-Service interface connects to the application layer with the purpose of sending all collected data to the users when requesting information is happened.

Application layer is the highest layer of the system that use to contact with the users by monitoring human behavior and control the system and show data that comes from the raw sensors context. To illustrate the information from raw sensor (hardware parts), application layer sends requests to the middleware and after that middleware will communicate to the lower layer as physical layer to bring containing data from sensors, then send back to the application layer and show to the user by API (Application Programming Interface). The hierarchical diagram of the middleware system is shown in Fig. 2.

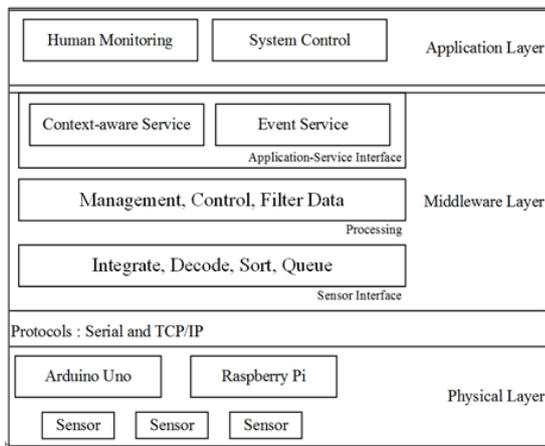


Fig. 2. Hierarchical diagram of the middleware layer.

B. Context-Aware Service

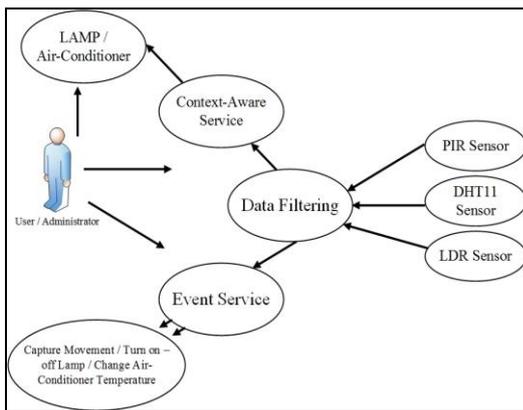


Fig. 3. Context-aware service.

Context Awareness relates of contextual information such as data filtering that bring up the data from the raw sensor and monitoring one’s behavior on the current context. The functional of the context-aware service as a control system of the smart laboratory room which control the facilities automatically namely lamps and air-conditioner from the motion sensors to achieve the goal of context awareness system, event processing service of the tasks perform as the common monitoring of the context as a Fig. 3 such as capturing movement, turn on/off the light, adjust the

air-conditioner automatically through the provided sensors that can input the real information such as location, time, temperature, the information can update by the system and user manually for communication with other devices and application.

C. Context-Aware Middleware Performance

```
if (n == 1) {
    sprintf (echoBuffer, "%s\n", "ON");
    recvMsgSize = strlen (echoBuffer);
} else {
    sprintf (echoBuffer, "%s\n", "OFF");
    recvMsgSize = strlen (echoBuffer);
}
```

Fig. 4. Input condition for status of presence.

```
memset(&echoServAddr, 0, sizeof(echoServAddr))
echoServAddr.sin_family=AF_INET;
echoServAddr.sin_addr.s_addr=htonl(INADDR_ANY);
echoServAddr.sin_port = htons(echoServPort)

if (bind(servSock, (struct sockaddr *) &echoServAddr,
sizeof(echoServAddr)) < 0)
    DieWithError("bind() failed");

if (listen(servSock, MAXPENDING) < 0)
    DieWithError("listen() failed");

for (;;)
{
    clntLen = sizeof(echoClntAddr);

    if ((clntSock = accept(servSock, (struct sockaddr *)
&echoClntAddr, &clntLen) < 0)
        DieWithError("accept() failed");
```

Fig. 5. TCP Echo Server connects to the client.

```
Random rand = new Random();

// FIRST DEVICE
n = 1;
msg =
sendAndWaitMsg(Integer.toString(n));
msg = msg.replaceAll("\r\n", "");
jsonObject.put("Value 1", msg);

// SECOND DEVICE
n = rand.nextInt(100) + 1; // input for
second device is randomized
msg =
sendAndWaitMsg(Integer.toString(n));
msg = msg.replaceAll("\r\n", "");
jsonObject.put("Value 2", msg);

// THIRD DEVICE
n = rand.nextInt(100) + 1; //input for
third device is randomized
msg =
sendAndWaitMsg(Integer.toString(n));
msg = msg.replaceAll("\r\n", "");
jsonObject.put("Value 3", msg);
String result = jsonObject.toString(); //
create result string

// send result to client (API)
return
Response.status(200).entity(result).build();
}
```

Fig. 6. Client that run on Tomcat Servlet from Apache.

According to the upper context-aware middleware architecture, we assumed that there are two systems which

made from 2 prototype platforms such as Raspberry Pi and Arduino Uno according to the status of the system that is shown in Fig. 4. These platforms perform as a mediator and read data from the raw sensor namely PIR sensors, Temperature and Humidity sensors, LDR sensors for monitoring surround environment.

For the Raspberry pi performs as a server itself is shown in Fig. 5 and we need to access via other side and implement function to help it perfectly to communicate with client side which bring data from the raw sensor to middleware layer and continue to display those data at the application program interface to the user.

Fig. 6 is used as a client to communicate with the server form Raspberry side. The output was taken directly from Raspberry pin status and others are randomized and echoed from middleware to raspberry pi server and back to middleware.

### V. RESULT AND DISCUSSION

We implemented the middleware for context aware system to be connector between physical layer and application layer. A middleware enables compound as a bridge that work together with protocols and platforms. The system receives the information from physical sector such as PIR sensor, Temperature and Humidity sensor, LDR sensor through prototype platforms, after that middleware will collect all data from raw sensor then will process randomly to share provided data to the application layer as well as displaying for the user.

Fig. 7 had shown the emulated from application program interface to the middleware, the output was taken directly from raspberry pin status, the other was randomized and echoed from middleware to raspberry server and then back to the middleware once more time.

A single data was displayed on the screen as a Fig. 8 is the information that shown the status of the input value that ON and OFF are the result from monitoring movement of the system.

To evaluate the performance of the proposed middleware system, the requirements of the middleware that we need to fulfill are essential role to implement and develop a better system. Graphic User Interfaces (GUIs) are used to show the monitoring of all provided sensors for middleware layer.

When we have started to operate the system all of sensor will send data to the middleware and collect to the database, at the meantime the graphics are showed the values of each sensors with different colors of patterns. On the other hand, different platforms are may receive a little different data according to the real environment.

The middleware would provide uniform abstraction and reliable services for common operations and would simplify the development of context aware application. It would be supported diversity to hardware, operating system and programming language. The middleware would also allow us to compose complex system based on the interaction between user and context aware application. More significantly, it would provide support for complex tasks such as contextual information which consisting in the hardware layer as well as raw sensors. It would also define a common model context that to ensure the different application in the smart environment. Furthermore, the middleware is supported the requirement such as mobility, heterogeneity and could handle with the complex system from hardware components for physical layer.

```

pi@raspberrypi ~/Documents/CSocket $ sudo gcc -o TCPEchoServer TCPEchoServer.c D
lewithError.c HandleTCPClient.c -lwiringPi
pi@raspberrypi ~/Documents/CSocket $ sudo ./TCPEchoServer 51717
Handling client 192.168.0.3
Data received : 1
Data from digital input Mode 1
Handling client 192.168.0.3
Data received : 57
Data from digital input Handling client 192.168.0.3
Data received : 84
Data from digital input Handling client 192.168.0.3
Data received : 1
Data from digital input Mode 1
Handling client 192.168.0.3
Data received : 65
Data from digital input Handling client 192.168.0.3
Data received : 91
{"Value 3":"91","Value 1":"OFF","Value 2":"65"}
    
```

Fig. 7. Raspberry interface running the process.

```

Handling client 192.168.0.3
Data received : 1
Data from digital input 1
Mode 1
{"Value 1":"ON"}

pi@raspberrypi ~/Documents/CSocket $ sudo ./TCPEchoServer 51717
Handling client 192.168.0.3
Data received : 1
Data from digital input 0
Mode 1
{"Value 1":"OFF"}
    
```

Fig. 8. Monitoring raspberry input and output.

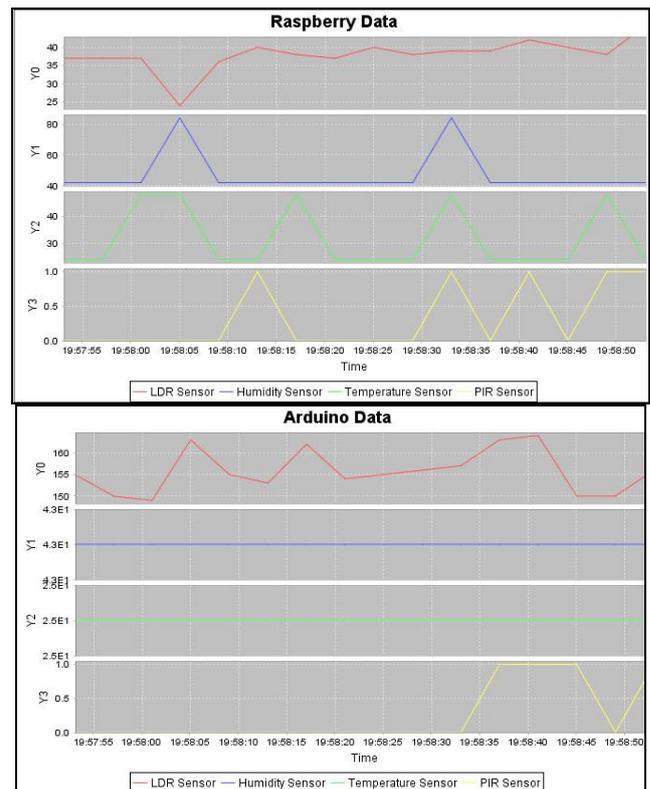


Fig. 9. Middleware user interface.

The Fig. 9 shows the value of the real data from multi-sensors such as LDR sensors, Humidity and Temperature sensors and PIR sensors that work with two different systems. The system collected the data with the real environment and provide those data to the database and be ready to show all of data to the user interface. This middleware could be managed and controlled the complexity of hardware layer and support to the application easily.

## VI. CONCLUSION AND FUTURE WORK

The context-aware middleware was developed to process data collected from multi-sensors through prototype platform that can solve the problem of dependence on hardware and application layers. The system could be combined services via the middleware of context aware system by connectivity of data from motion sensor. The operability and scalability of the middleware was maximized by adding intelligent context aware, event service and context functions. Middleware and application services make use of different level of contexts and adapt the direction which behave depending on current context as the movement of human or behavior that performed to the system. To perform the proposed middleware architecture we applied and implemented integration the system of control management and context awareness together to make them work perfectly and response to the user as a real time responding between middleware to application and interfaces to the user.

Future research should to implement the system which can be work with the both computer and mobile computing bases, web application should be applied to show the information from the middleware and show the graph of value between the system consisting of middleware and non-middleware system.

## ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisors, colleagues for their support and help on all aspects from all technical to material and I would also like to appreciate my scholarship JICA Project for AUN SEED-NET (ASEAN University Network, Southeast Asia Engineering Education Development Network) for their financial support on this research.

## REFERENCES

[1] K. E. Ek Kjaer, "A survey of context-aware middleware," in *Proc. the 25th conference on IASTED International Multi-Conference: Software Engineering*, 2007, pp. 148–155.

[2] W. Liu, X. Li, and D. Huang, "A survey on context awareness," in *Proc. 2011 International Conference on Computer Science and Service System (CSSS)*, 2011, pp. 144–147.

[3] J.-W. Chang and H.-J. Lee, "Context-aware architecture for intelligent application services in ubiquitous computing," in *Proc. 2007. ICSC 2007. International Conference on Semantic Computing*, 2007, pp. 275–281.

[4] T. Mo, W. Li, W. Chu, and Z. Wu, "CABS3: Context-awareness based smart service system," in *Proc. 2010 6th International Conference on*

*Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010, pp. 1–4.

[5] J. Hong, E. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Syst. Appl.*, vol. 36, pp. 8509–8522, 2009.

[6] Y. Bai, H. Ji, Q. Han, J. Huang, and D. Qian, "MidCASE: A service oriented middleware enabling context awareness for smart environment," in *Proc. International Conference on Multimedia and Ubiquitous Engineering*, 2007, pp. 946–951.

[7] J. Viterbo, V. Sacramento, R. Rocha, G. Baptista, M. Malcher, and M. Endler, "A middleware architecture for context-aware and location-based mobile applications," in *Proc. 32nd Annual IEEE Software Engineering Workshop*, 2008, pp. 52–61.

[8] H. K. Pung, T. Gu, W. Xue, P. P. Palmes, J. Zhu, W. L. Ng, C. W. Tang, and N. H. Chung, "Context-aware middleware for pervasive elderly homecare," *IEEE J. on Sel. Areas Commun.*, vol. 27, no. 4, pp. 510–524, 2009.

[9] C. G. Sahu and D. Adane, "A survey on context-aware middleware."

[10] H.-X. Ji, Y.-B. Bai, G.-L. Dai, J.-H. Li, J. Huang, and D.-P. Qian, "MidCASE: Service-oriented middleware prototype enabling context-awareness in wireless sensor networks," *Appl. Res. Comput.*, vol. 11, p. 075, 2007.

[11] Z. Aziz, C. Anumba, D. Ruikar, P. Carrillo, and D. Bouchlaghem, *Semantic Web Based Services for Intelligent Mobile Construction Collaboration*, ITcon, 2004.

[12] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *J. Netw. Comput. Appl.*, vol. 28, no. 1, pp. 1–18, 2005.

[13] P. Fahy and S. Clarke, "CASS—a middleware for mobile context-aware applications," *Workshop on Context Awareness, MobiSys*, 2004.



**Khamla Non-Alinsavath** was born in Xiengkhuang province, Laos, on May 15, 1989. He has received a bachelor of engineering, majoring in information technology at the Faculty of Engineering, National University of Laos, Vientiane Capital, Lao PDR in the Academic year 2010-2011.

Currently he is a master student as well as researcher at Electrical Engineering and Information Technology Department, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia. His areas of interests are computer network, wireless sensor network, context awareness, and context-aware middleware.



**Lukito Edi Nugroho** is currently working as vice dean of Faculty of Engineering, Universitas Gadjah Mada. He was Ph. D candidate at School of Computer Science and Software Engineering, Monash University, Australia (2000), the master, James Cook University of North Queensland, Australia (1992), Undergraduate, University Gadjah Mada, Indonesia (1989). His areas of interest are context-aware system, mobile computing, software engineering, and information system.



**Agus Bejo** respectively received his B.Eng., M.Eng, and D.Eng. degrees from the Department of Electrical Engineering, Universitas Gadjah Mada, Indonesia in 2003, the Department of Electrical Engineering, Chulalongkorn University, Thailand in 2007 and the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Japan in 2014. He is currently a lecturer in the Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Indonesia. His research is focused on the processor architecture and compiler design, system-on-chip and fingerprint authentication applications.