Integration of multilayered context-aware control system for ubiquitous computing environment

Khamla Non Alinsavath1*, Lukito Edi Nugroho2 and Widyawan3
1, 2, 3 Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia.

Email: khamla@mail.ugm.ac.id1, lukito@ugm.ac.id2, widyawan@ugm.ac.id3

Abstract: In response to recent demand, context-aware application performs significantly in the field of pervasive computing and smart environment. Many approaches have been proposed to provide functionalities and implementation using context information to observe the environment. Developing context-awareness system becomes more essential issue in order to support into the real situation and system deals with many components and complexity. There are many hardware with different platforms that make them have difficulty to communicate to each other with different layers processing in the system of context such as physical, middleware and application layers. Services in each layer need to adapt the way they behave according to the current context information. This research introduces the development of context-aware system and middleware developments that support the creation of smart application with the purpose of providing the framework of the system and deploy middleware as being the bridge to integrate physical layer and application layer seamlessly to support monitoring ubiquity environment. The system controls multi sensors to monitor the real environment from raw data through prototype platforms for managing, sensing, computing and user interacting as the outcome of the system’s performance. Implementation and experiment have been performed where user moves in / moves out from the sensor range. The proposed middleware is able to handle the complicated system and fulfill its requirement such as heterogeneity, scalability and mobility, the system and complicated components are integrated to be a completed system that operate, interpret and representation context entities to an end users. Having this control system makes a lot of benefits such as real time information access, become intermediary between layers and support to control energy system with the purpose of reducing the energy consumption, rapid communication to the users.

1. Introduction
Context awareness [1] is an important part of pervasive computing field to monitor and control human behavior and real environment. A system can lead to the development of a rich knowledge-based that will not only be useful for society, but also serve as an essential input for a system’s processing. The definition of the context is from Schilit Adams [2] which introduce the concept of context is related to the location, nearby person, hosts or objects as well as changes of them over the time context. Context [3] is defined as situational information of entities (place, object, or people) [4], the system has to acquire context correctly and provide adequate service behaviors regarding to provided context. Context is divided into three categories: computing context, user context, physical context. In order to achieve better understanding more about context, Chen and Kots [5] add time context such as time of a...
day, week, month and season of the year. Time information like a context’s history that adds to this definition.

Context awareness is the ability of a program or computing device to send and act upon the information in its environment such as location, time and user identities [6][7]. In the real environment, context changes dynamically [8] and the applications have to adapt their behavior, therefore context-aware system need to manage its context information effectively and satisfy qualities of context information required by context-aware applications [9]. The real-time capability of new human behavior monitoring solutions can provide mission critical artificial intelligence [10]. Hence computing infrastructure for understanding human movement processing is increasingly important, especially when merging this computing infrastructure and technology in order to obtain more advantage and efficiency while processing data with innovation, transformative methodology and successfully solutions. Context awareness [11][12] also can be broadly classified into the following areas: self-contains and infrastructure based context awareness, context information is important to apply to the system and situation awareness [13] such as location, identity, time and activity that follow with these questions of 5Ws: who, what, then, where and why, which is minimum necessary set of context information in the pervasive computing environment.

This research aims to develop context-aware application to deal with existing issue, to understand user events and integrate the system into seamless application by middleware that can support and being intermediary between physical and application layers without obstacle, it is frequently protected by more efficient monitoring techniques. Middleware can be a solution for a complicated system to access data easily and comfortably, and also to fulfill the requirement of context-aware middleware such as heterogeneity, mobility and ease of deployment. On the other hand, context-aware application has been proposed to implement together with middleware and become application program interface to read data from raw sensors through middleware and to monitor the environment conditions.

The remainder of this paper is organized as follows: section II provides background and related works on applying context-awareness scenarios and ubiquitous computing. Section III introduces multilayered context-aware system architecture including hardware, middleware, and application frameworks. In the section IV, the performance of the experiment, result and discussion are presented. Section V concludes the entire work of this paper and future work for the paper.

2. Related Works
The context-aware system [14] as the crucial component in Ubiquitous Computing [15], perceives context changes and adapts accordingly. It provides an important way to improve intelligence system, and comes up with a convenient and efficient interaction method between the intelligence system and the surrounding environment by middleware. The context-aware system has a general architecture divided into three tiers: context-aware application, context-aware service, and context sensors. Many researchers have proposed different context aware middleware schemes to overcome different problems. To find a simple, fast and qualified algorithm for middleware application based on context awareness is a scenario widely studied by a lot of researchers. Limon et.al [16] presented a context aware middleware architecture which provides support for both pervasive and ad hoc computing. The middleware is structured in terms of a number of component frameworks, for example, the middleware platform provides support for dissemination of events in a timely fashion. Mechanism for context awareness and intelligent decision-making are also provided. Lastly a flexible framework is offered for enabling the use of a number of devices service discovery protocols. Bai et.al [17] presented the middleware to support the application development of context awareness under the new-type networks environment represented by wireless sensor network. It applied the updated service-oriented and light-weight structure with excellent expansibility and efficiency in the running process. Additionally, the architecture of middleware mainly contains five layers and cross two layer modules: hardware abstract, service registry, context model, reasoning and application representation layers; energy management module and security module. During the process of context awareness, the new method of awareness synchronization is designed to ensure sensitivity to context switch.
3. Multilayered Context-Aware System Architecture and its application

3.1. Multilayered Context-Aware System Design

We present the design and architecture of the context-aware system which serves the platform of desktop applications. It provides a mean to store and retrieve contextual information, and also facilitates providing relevant services to the application so that the context information can be used more effectively. The system communicates with the database and other service for gathering context information and then present it to the client applications. This proposed application is also designed to have advanced context usage functionalities such as learning from human behaviors, weather and light intensity. Additionally, entities such as people, objects and location would themselves serve as contextual information sources. The system is designed to aid a human decision maker and help people that interact with a user interface, and the designed architecture has essential features of context-aware systems namely managing, controlling, filtering data and intermediary between layers.

Most of the proposed systems are user centric and the context of a situation is defined only with respect to the entities involved in the events, activities, properties and relationships between them. To tackle these drawbacks, we have developed the context-aware middleware that is being designed to support the requirements of the middleware in the pervasive environment such as the proposed system can support for heterogeneity, scalability, mobility and so on. It supports integration of different application and collection and storage information from heterogeneous services.

Figure 1 shows that the context-aware system consists of three layers such as the application layer, middleware layer and physical layer. The details of each layer are listed below:

The physical layer consists of four kind of sensors namely Passive Infrared sensors, light-dependent resistor and Temperature and Humidity sensors, they are used to capture and communicate with the real environment through prototype platforms. The middleware layer performs as a bridge between physical and application layers and being integration point for the system seamlessly. After receiving data from the hardware then the middleware will manage and control the data and collect data from the physical layer and data will be recorded to the database. The application layer is the top layer of the proposed system that is used to communicate with the users by monitoring human activity, controlling the system and providing data when users request data through the system from sensors.

3.2. Proposed Context-aware middleware for the system

The middleware provides an abstraction and reliable services for common operations and would simplify the development of context aware applications. It would be accommodated a variety to hardware, operating systems and programming languages. The middleware would also allow us to compose complexity of the systems based on the interaction between users and context aware applications. More significantly, it supports for complex concern such as contextual information in the hardware layer as well as raw sensors. It is also defined a common model context that to ensure the different application in the smart environment.
Figure 2: Proposed Context-aware middleware architecture

The proposed middleware specifies certain requirements for middleware’s in the smart environment, for which terms of requirement are as follows:
1. The proposed middleware should support collection of context information from heterogeneous sensors and services and interaction of appropriate context information from real conditions.
2. It should support the mobility of the system uses and efficiency interface to monitor the system.
3. The application layer can easily access data from the middleware layer, it simply make a request to the middleware and then middleware will manage then deliver the requested data to the application.
4. It should handle complexity of the system and different contexts between application and services.

3.3. Context-aware service and application processing
The context-aware service diagram that illustrated in figure 3 collects information about the current environment which receives from the real sensors and sends received data such as human movement, temperature, light intensity and humidity to the system to filter and analyze as real time services. The information can be updated by the user with the application and raw sensors communicate to each other through the proposed middleware.

Context–aware services adapt to the environment of the user automatically and the system allows for interactions between user and application by different kinds of sensors that control the light bulb and air-conditioner, along with human activities such as moving in - out the sensor range and light intensity.

Figure 3: Context-Aware service diagram

The figure 3 presents the processing of the application which works properly according to all components. All of the data from the sensors captured from real conditions are sent to middleware through the TCP/IP protocol and the Serial protocol, after that the collected data are sent from middleware to the database.

4. Result and Discussion
4.1. Experimental Design
The experiment starts from the physical layer, then moves through the middleware layer and finishes with application layer. We have implemented an equipment setup that consists of many devices such as sensors, prototype platforms, and personal computer which are connected by serial and Ethernet cables. The whole system is illustrated in the figure 4.

Figure 4: Equipment Setup of the Experiment

The result of the proposed system can appropriately capture the conditions and situation. If situation detected is mean that there is someone moving within the sensor range. As a result, the LED (light-emitting diode) will turn on. The LED will stay on as long as the people move around, and after a period where no movement is detected, the LED will turn off. The situation is depended on data from middleware layer which come from real detection of the environment. The middleware layer deals with two different systems and eight sensors (four sensors for Arduino platform, four sensors for Raspberry platform) and collects all of the real data from the physical layer. Moreover the middleware is responsible for arranging the required data when users request. Sensors provide data about events in the real situation and classify the situation into context such as time context, location context and presence context after collecting all of data then transferring them to the middleware layer.

4.2. Middleware layer implementation and integration point
To evaluate the performance of the proposed middleware system, the requirements of the middleware that we need to fulfill are essential to implementing and developing a better system. This middleware layer becomes intermediary between physical and application layers, also it integrates whole system to be seamlessly application with the completion of service and provide raw data to the users through this middleware layer. After the function is completed and able to monitor the system from the middleware layer, all of raw data from the sensor are stored in the database (table 1). The latest data received from multi sensors will be stored for one month and then will be replace new data automatically. The middleware layer will receive all of the services from the hardware layer that work to completely success and cover the complexity of physical services. All of collected data are then ready to be sent to the application layer. In order to connect with other processes, we need to have other packages or services to communicate with sensors by using protocols such as the Serial protocol which is used to communicate with Arduino and multi-sensors, or the TCP/IP protocol, which is used to communicate with Raspberry and multi-sensors.

Table 1: Database for collecting data from monitoring environment

| ID  | Arduino_LDR | Arduino_Humid | Arduino_Temp | Arduino_PIR | Raspberry_LDR | Raspberry_Humid | Raspberry_Temp | Raspberry_PIR |
|-----|--------------|----------------|--------------|-------------|---------------|-----------------|----------------|---------------|
| 876 | 26           | 47             | 25           | 0           | 97            | 49              | 25             | 0             |
| 877 | 25           | 46             | 17           | 0           | 93            | 49              | 25             | 0             |
| 878 | 24           | 46             | 25           | 0           | 91            | 49              | 25             | 0             |
| 879 | 25           | 46             | 25           | 0           | 89            | 49              | 25             | 0             |
| 890 | 25           | 46             | 25           | 0           | 96            | 49              | 25             | 0             |
| 891 | 26           | 46             | 50           | 0           | 95            | 49              | 25             | 0             |
The table 1 shows the latest data received from the multi-sensors that will stored for one month. The data consists of 2 different systems and 6 values of data from the sensors such as LDR sensor, Humidity and Temperature sensors and PIR sensors.

**4.3. Application Layer implementation and result**

We have implemented the application layer called the context-aware application to monitor and control the whole system. There are two kind of interfaces that we have implemented such as package data that shows one hundred data from database that bring from middleware which illustrated in the table 3 and single data that shows the latest data from real environment in the table 4. The application will directly communicate with the middleware, after which the middleware will handle the physical layer and obtain data from sensors to illustrate in the provided database and prepare to show in the API interface. The proposed middleware is able to deal with the complexity of the system in the hardware components that makes it easier for users to use and operate the completed system. Middleware hides the complexity of the hardware layer so that the application could access the middleware anytime when user inquires.

| Table 2: Assigned data to illustrate on application layer |
|-----------------------------------------------------------|
| jsonObject.put("raspberry_LDR", dt[0]);                |
| jsonObject.put("raspberry_humidity", dt[1]);          |
| jsonObject.put("raspberry_temperature", dt[2]);       |
| jsonObject.put("raspberry_PIR", dt[3]);               |
| jsonObject.put("arduino_LDR", dt[4]);                 |
| jsonObject.put("arduino_humidity", dt[5]);            |
| jsonObject.put("arduino_temperature", dt[6]);        |
| jsonObject.put("arduino_PIR", dt[7]);                 |
| String result = jsonObject.toString();                 |

Users interact with the application layer directly from the application through web browser. The application layer implements a communicating component and the availability of all communication between applications requires cooperation managed by the system and application layer. The end-user processes are supported by this layer, the quality of service is identified, and user authentication and security are considered. Moreover to make sure that effective communication with another layer can operate properly, we provided the services to ensure that the application can contact the database completely and process the data for the user by using internet HTTP protocols.

| Table 3: Sensor Data are shown on application programming interface |
|---------------------------------------------------------------|
| Number | LDR  | Humidity | Temperature | PIR  | LDR  | Humidity | Temperature | PIR |
|--------|------|----------|-------------|------|------|----------|-------------|-----|
| 1      | 16.50| 40.00    | 25.00       | 0.00 | 13.00| 40.00    | 25.00       | 0.00|
| 2      | 17.30| 40.00    | 35.00       | 0.00 | 13.00| 40.00    | 35.00       | 0.00|
| 3      | 20.40| 40.00    | 25.00       | 0.00 | 13.00| 40.00    | 25.00       | 0.00|
| 4      | 19.00| 40.00    | 35.00       | 0.00 | 13.00| 40.00    | 35.00       | 0.00|
| 5      | 19.40| 40.00    | 50.00       | 0.00 | 13.00| 40.00    | 50.00       | 0.00|
| 6      | 15.40| 40.00    | 25.00       | 0.00 | 13.00| 40.00    | 25.00       | 0.00|
| 7      | 20.70| 40.00    | 35.00       | 0.00 | 13.00| 40.00    | 35.00       | 0.00|
| 8      | 19.40| 40.00    | 50.00       | 0.00 | 13.00| 40.00    | 50.00       | 0.00|
| 9      | 20.00| 40.00    | 25.00       | 0.00 | 13.00| 40.00    | 25.00       | 0.00|
| 10     | 17.30| 40.00    | 50.00       | 0.00 | 13.00| 40.00    | 50.00       | 0.00|

**Table 4: Monitoring Application Program Interface Single data as the latest data**

```
(jsonObject.put("raspberry_humidity", 44);
(jsonObject.put("arduino_humidity", 46);
(jsonObject.put("arduino_temperature", 184.5);
(jsonObject.put("raspberry_PIR", 6);
(jsonObject.put("arduino_PIR", 0);
(jsonObject.put("raspberry_temperature", 125);
(jsonObject.put("arduino_LDR", 143);
```
5. Conclusion
The area of the context-aware system and integration approach are provided in this research. The research touches many subjected such as sensing, control designing, integration layers, application development, et cetera. The main focus lied on applying middleware for being the bridge between physical and application layers, it has been developed to process data collected from provided sensors through prototype platforms that can be solved the problem of dependent on hardware and application. We applied and implemented integration system between the control system and context-aware computing to make them work together properly and respond to the user in real time from lower layer to application layer. An architecture of the system efficiently supports and fulfills the requirement of the system, represent and integrate contextual information, and act as the intermediary where different layers and complexities can be recovered by the proposed system. All layers are combined seamlessly and performs as completed system when they send data from raw sensor through proposed system with ease when user request data from system.

Acknowledgment
I would like to express my deeply acknowledge to those who assisted, inspired and supported me during the completion of this research such as my supervisors, colleagues. Financial support from the AUN/SEED-Net JICA project in this research and scholarship is greatly appreciated.

References
[1] Ö. Yürür, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung, “Context-Awareness for Mobile Sensing: A Survey and Future Directions,” IEEE Commun. Surv. Tutor., vol. 18, no. 1, pp. 68–93, Firstquarter 2016.
[2] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications,” in Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on, 1994, pp. 85–90.
[3] 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.
[4] S. Lee, S. Park, and S. Lee, “A Study on Issues in Context-Aware Systems Based on a Survey and Service Scenarios,” in 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009, pp. 8–13.
[5] G. Chen, D. Kotz, and others, “A survey of context-aware mobile computing research.”
[6] W. Liu, X. Li, and D. Huang, “A survey on context awareness,” in Computer Science and Service System (CSSS), 2011 International Conference on, 2011, pp. 144–147.
[7] K. B. Balavalad, S. S. Manvi, and A. V. Sutagundar, “Context Aware Computing in Wireless Sensor Networks,” in 2009 International Conference on Advances in Recent Technologies in Communication and Computing, 2009, pp. 514–516.
[8] T. Mo, W. Li, W. Chu, and Z. Wu, “CABS3: Context-Awareness based smart service system,” in Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, 2010, pp. 1–4.
[9] A. Wood et al., “Context-aware wireless sensor networks for assisted living and residential monitoring,” Netw. IEEE, vol. 22, no. 4, pp. 26–33, Jul. 2008.
[10] F. Gui, N. Zong, and M. Adjouadi, “Artificial Intelligence Approach of Context-Awareness Architecture for Mobile Computing,” in Intelligent Systems Design and Applications, 2006. ISDA ’06. Sixth International Conference on, 2006, vol. 2, pp. 527–533.
[11] E. Kim and J. Choi, “A Context Management System for Supporting Context-Aware Applications,” in Embedded and Ubiquitous Computing, 2008. EUC ’08. IEEE/IFIP International Conference on, 2008, vol. 2, pp. 577–582.
[12] J.-W. Chang and H.-J. Lee, “Context-aware architecture for intelligent application services in ubiquitous computing,” in Semantic Computing, 2007. ICSC 2007. International Conference on, 2007, pp. 275–281.
[13] K.-E. Ko and K.-B. Sim, “Development of context aware system based on Bayesian network driven context reasoning method and ontology context modeling,” in 2008 International Conference on Control, Automation and Systems, 2008, pp. 2309–2313.
[14] H. Mcheick, “Ubiquitous Computing and Context-aware Applications: Survey and Contributions,” in 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016, pp. 394–397.

[15] Z. A. Hua, “Study of ubiquitous learning environment based on Ubiquitous computing,” in 2010 3rd IEEE International Conference on Ubi-Media Computing, 2010, pp. 136–138.

[16] H. A. Duran-Limon et al., “Context-aware middleware for pervasive and ad hoc environments,” Context Tech Rep, 2003.

[17] Y. Raja Vara Prasad and P. Rajalakshmi, “Context aware building energy management system with heterogeneous wireless network architecture,” in Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP, 2013, pp. 1–8.