International Journal of Smart Home
Vol. 9, No. 8 (2015), pp. 169-178
http://dx.doi.org/10.14257/ijsh.2015.9.8.18

Context-Aware Computing for Delivering u-Healthcare Services

Yvette E. Gelogo[1], Haeng-Kon Kim[1*] and Rhan Jung[**]

[1]School of Information Technology, Catholic University of Daegu, Korea
[2]Department of Computer Science, Kang Won National University, Korea
[1]yvette@cu.ac.kr, [1]hangkon@cu.ac.kr and jungran@kangwon.ac.kr

Abstract

Ubiquitous healthcare systems will monitor patients as they maintain their normal everyday activities, in order to warn the patients or healthcare workers of problems as well as collecting data for trend analysis and medical research. A healthcare application may bring about severe network traffic as patients, biomedical signals, sampling rates increase. This traffic could produce some end-to-end delay. A compression of sensing data may be a solution, but it can also cause an increment of processing time and unexpected errors in healthcare service. In this paper we proposed a component based development framework for context-aware processing for u-healthcare application that can reduce network traffic, by using the framework, the information gathered from the environment and patient condition will be process using the designed component development and send to the user or hospital in form of different services (1) alert message service, (2) location information services (3) condition of patient services. We also present the sensor interface framework and multi-purpose gateway to process the contextual data and send to medical centers, hospital or patients mobile device as u-healthcare services.

Keywords: *context-aware systems, u-healthcare, component based development*

1. Introduction

Ubiquitous healthcare systems will monitor patients as they maintain their normal everyday activities, in order to warn the patients or healthcare workers of problems as well as collecting data for trend analysis and medical research. The continuous monitoring of the health record would give a better diagnosis, treatment and emergency services.

The ubiquitous computing needs many technologies such as network, application, software platform, and device standardization. Users should be able to access the ubiquitous applications conveniently as well as safely. The healthcare service out of ubiquitous applications may provide ubiquitous medical service for human beings. If component technologies are developed, healthcare will be improved. An advance in various sensors is worthy of notice. There are many examples about the biomedical sensors such as blood ingredient detection sensor, environment monitoring sensor, body signal analysis, and activity tracing technologies. There is healthcare applications based on sensor network that can notify the doctor of urgent messages of heartbeat, oxygen saturation, and electrocardiogram (ECG) or monitor health status through badges attached on their clothes [1][3].

The monitoring sensors or devices for a living body must automatically send warning signals to clinic team whenever some events arise. A healthcare application may bring about severe network traffic as patients, biomedical signals, sampling rates increase. This traffic could produce some end-to-end delay. A compression of sensing data may be a solution, but it can also cause an increment of processing time and unexpected errors in

[*] Corresponding Author

ISSN: 1975-4094 IJSH

healthcare service. This end-to-end performance in u-healthcare can be improved by transmitting differential signals or changes since the last transmission. Another way is to use the current context of the patient. For example, some trivial vital signs can be skipped by a medical information application. Also this context recognition could let clinic team provide better treatment for the patient [1][3]. To address these limitations, we proposed a component based development framework to process the context data from the patient and process these context to for, a u-healthcare services (a) alert messages, (b) location information (c) condition of the patient.

2. Background of the Study

2.1. Definition of "Context"

- Computing context, such as network connectivity, communication costs, and communication bandwidth, and nearby resources such as printers, displays, and workstations.
- User context, such as the user's profile, location, people nearby, even the current social situation.
- Physical context, such as lighting, noise levels, traffic conditions, and temperature.
- Time context, such as time of a day, week, month, and season of the year.

2. 2. Context-aware Computing

Although information about the current context may be available to mobile applications, how to effectively use that information is still a challenging problem for application programmers. Schilit defines context-aware computing by categorizing context-aware applications as follows [1]:

- *Proximate selection,* a user-interface technique where the objects located nearby are emphasized or otherwise made easier to choose.
- *Automatic contextual reconfiguration,* a process of adding new components, removing existing components, or altering the connections between components due to context changes.
- *Contextual information and commands,* which can produce different results according to the context in which they are issued.
- *Context-triggered actions,* simple IF-THEN rules used to specify how context-aware systems should adapt.

While the above categorization identifies classes of context-aware applications, Pascoe [2] proposes taxonomy of context-aware features, including contextual sensing, contextual adaptation, contextual resource discovery and contextual augmentation (the ability to associate digital data with a user's context). Dey combines these ideas and maps them to three general categories of context-aware features that context-aware applications may support: presentation of information and services to a user, automatic execution of a service, and tagging of context to information for later retrieval [3].

From our definition of context in the previous section, however, we have a different perspective on how a mobile application can take advantage of context. There are essentially two ways to use context: automatically adapt the behaviors according to discovered context (using active context), or present the context to the user on the fly and/or store the context for the user to retrieve later (using passive context). Thus we give two definitions of context-aware computing:

- *Active context awareness:* an application automatically adapts to discovered context, by changing the application's behavior.

- *Passive context awareness:* an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.

Context-aware computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people and devices, and user activity). Since it was proposed about a decade ago, many researchers have studied this topic and built several context-aware applications to demonstrate the usefulness of this new technology. Context-aware applications (or the system infrastructure to support them), however, have never been widely available to everyday users.

2.2.1. Context-aware Application in u-healthcare

Context-aware mobile agents are a best suited host implementing any context-aware applications. Modern integrated voice and data communications equips the hospital staff with smart phones to communicate vocally with each other, but preferably to look up the next task to be executed and to capture the next report to be noted.

However, all attempts to support staff with such approaches are hampered till failure of acceptance with the need to look up upon a new event for patient identities, order lists and work schedules. Hence a well suited solution has to get rid of such manual interaction with a tiny screen and therefore serves the user with

- automated identifying actual patient and local environment upon approach,
- automated recording the events with coming to and leaving off the actual patient,
- automated presentation of the orders or service due on the current location and with
- Supported documenting the required information keying in a minimum of data into prepared form entries.

2.3. Component Based Development

The active repository is capable of providing information to the users by monitoring their development activities without the need to receive explicit queries from them. These systems execute in background inside an integrated development environment (IDE) monitoring the user activity and suggesting possible software components to be used in the current development context. Both the traditional and the active repositories provide functionalities only during the development phase of the software system. Once the system is deployed, the repositories are of no use.[8]

This way, in order to promote the runtime adaptation of context-aware component-based systems, this paper presents an active repository that actively provides new software components that are more adapted to the context of the adaptable system. Since the repository has a wide knowledge of the available components, it can better decide which configurations of components are more suitable to the running system.

In the proposed approach, the context aware system informs to the repository its current configuration and its context information and the repository is able to compute the components and new architecture that better fit the given context. In this way, the component repository commonly adopted in component-based software systems is expanded to provide components, not only during the development stage but also context aware components during the operational stage of the system life cycle.

3. Convergence Architecture for u-Healthcare Services

3.1. WBAN and WLAN for U-Healthcare

Figure 1 show the configuration of the u-healthcare convergence network which is built by integrating WBAN and WLAN. The WBAN is used for continuous monitoring of various vital signs such as EEG, ECG, BP, etc. It can also be used for transmitting multimedia data from MP3 devices and video cameras. In the cost-effective u-healthcare network systems, vital signs which are collected from noninvasive physiological sensors are first transported wirelessly through WBAN, and then through WLAN, and finally through the Internet to the medical information servers and/or doctors, using fixed-mobile convergence technology. In addition to the vital signals, the contextual information such as weather, temperature and emergency case may be transported for context-aware patient-specific real-time treatments.

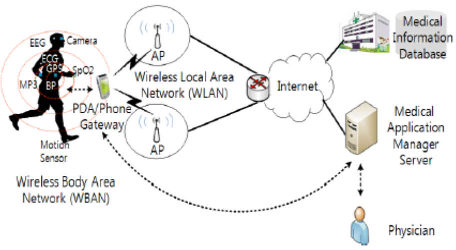

Figure 1. Configuration of WBAN and WLAN Convergence Network for U-Healthcare Service

The U-Healthcare Multi-Purpose Gateway in Figure 2 performs the convergence function for the integration of WBAN and WLAN. Both patients' data and contextual information are integrated at the gateway, and transported to the medical centers or hospitals using IP protocol over the Internet. This convergence gateway can be realized by a smart phone or other mobile devices such as a tiny PC and PMP. The main convergence function of the mobile gateway includes the support of interoperability of WBAN and WLAN, and the seamless mobility management between WLANs.

The IEEE has recently started to develop a standard for the wireless body area network (WBAN) [4]. The IEEE

802.15 Task Group 6 (BAN) [4] is developing a communication standard optimized for low power devices and operation on, in or around the human body (but not limited to humans) to serve a variety of applications including medical, consumer electronics / personal entertainment and so on. The characteristic of WBAN is that it can not only provide the transport service of the medical physiological information such as ECG, EEG and SpO2, but also provide the services for high-bandwidth multimedia services.

3.2. Sensor and Interface Framework for Context Gathering

Basically, UPnP sensor is a type of software module which presents the UPnP device for interconnection. This sensor framework controls the unloaded devices and the present UPnP devices connected to load and translate the protocol. Namely, it is likely to be an emulator of UPnP devices although non-UPnP in reality. We propose the system architectures to be design and implement.

The bio-sensor and environment sensor module being connected through UPnP sensor framework to Zigbee network which connect UPnP stacks. This sensor framework recognizes several sensor modules through UPnP middle-ware which has environment and bio-modules. This framework activates different user control and UPnP devices based on DHCP servers. To provide connectivity with utility flexibility and standards UPnP should be constructed with TCP/IP based UPnP stack through UPnP middle-ware. Because sensor module is not connected with non-IP device, therefore, the UPnP sensor device module inside the framework was constructed to form a virtual UPnP device.

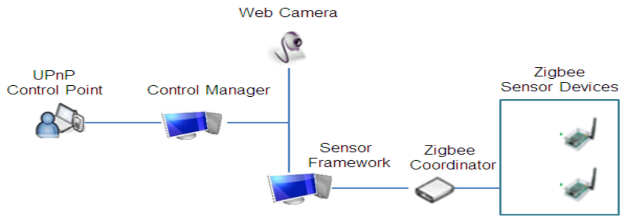

Figure 2. Sensor and Interface Framework for Receiving Context Data

3.3. Multi-Purpose Gateway Interface Environment

The communication protocol between PHDs (Personal Healthcare Device), CE (Computing Engine) and MS (Monitoring Server) are very important. With this, we designed a multi-purpose gateway using ISO/IEEE 11073 between PHDs and CE modeled by design preferences. The PHDs send message to CE one at a time is real application. However, CE should be capable of receiving multiple messages from PHDs. Therefore, we design a multi-purpose gateway that can handle multiple message transmission from PHDs. The challenge is to work on the efficiency of the transmission and design preferences are applied in order to map it dynamically. Figure 3 shows the multi-purpose gateway interface environment.

In this paper we also consider the communication between the CE and MS. IEEE 805.15.6 is the applied transmission protocol between CE and MS for WBAN network.

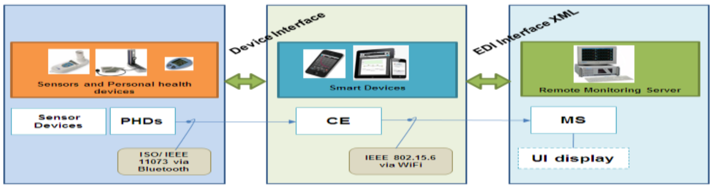

Figure 3. Multi-Purpose Gateway Interface Environment

The interface environment is characterized as Figure 3. It shows Device interface between the PHDs and the gateway, also it shows the interface between the gateway and EDI. IEEE P11073-20601 which compose data exchange standard on application layer define logical session, data transmission, session disconnection on the view point of object.

4. U-healthcare Model for Context-aware

4.1. U-Healthcare Prototype

Ubiquitous healthcare is an emerging technology that promises increases in efficiency, accuracy and availability of medical treatment. The purpose of u-healthcare monitoring system is to provide convenient healthcare service to both caregivers and patients, and to make it easy to diagnose patient's health condition. People can monitor their health without visiting the hospital or clinic.

Ubiquitous healthcare systems will monitor patients as they maintain their normal everyday activities, in order to warn the patients or healthcare workers of problems as well as collecting data for trend analysis and medical research. The continuous monitoring of the health record would give a better diagnosis. The integration of wireless body sensor network is very important for timely monitoring.

The information from the sensors, like heart rate, vital signs, and location of the patient is being monitored. The context information is gathered using mobile devices and this context information will be processed. Figure 4 shows the u-healthcare prototype with context-aware application.

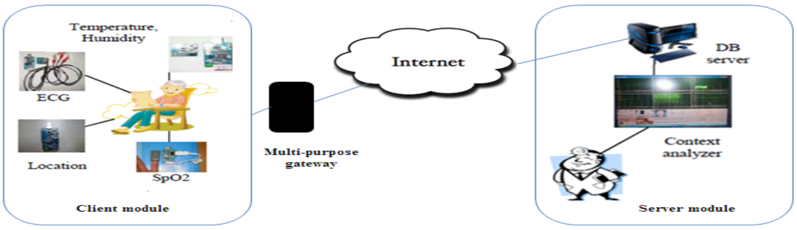

Figure 4. U-healthcare prototype with context-aware application

4.2. Context-Aware Computing Application for U-Healthcare

4.2.1. System Architecture

In this study we have designed the CBD-IF (CBD-Intelligent Framework) framework. This framework is divided into two modules, the client and server modules that will handle the client system context changes and generate new system architectures over the time through component reconfiguration. Figure 5 shows CBD-IF's general architecture. This framework is not intended to be restricted to a specific component model. However, to use CBD-IF's services, the client system needs to be developed using a component model that allows at least the simple reconfiguration commands (add, remove, replace, connect, or disconnect a component in the system).

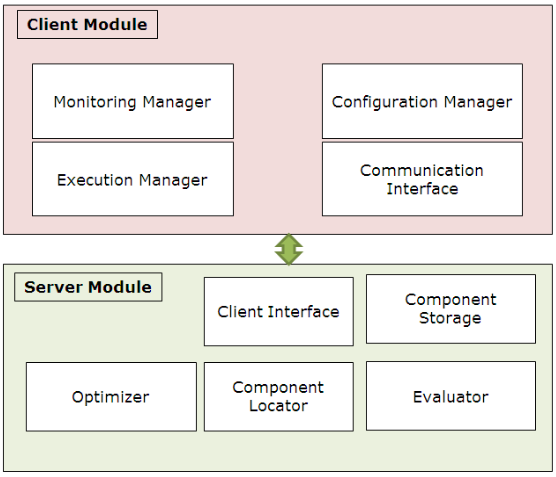

Figure 5. CBD-IF's General Architecture

The CBD-IF framework is divided into two disjoint groups, the client module and the server module. As shown in Figure 5, the Client components are as follows.

A. Monitoring manager: The monitoring manager is responsible for monitoring the environment, if there is a presence of new context, the change o user's location, the available components based on the received context.

B. Execution Manager: responsible to combine information collected by the Monitoring Manager in a meta-representation that will be used to generate a new optimized configuration to the system.

C. Configuration Manager: This is responsible for reading the new context-aware information received from the monitoring manager and instantiate new architectures sent by the CBD-IF server, new architectures will be instantiated by adding, removing, or replacing components or interconnecting existing ones.

D. Communication Interface: provides communication with CBD-IF server. This module is able to provide the client system an abstraction of the server location in order to enable the use of the distributed services.

The CBD-IF server provides a set of services for receive client system context and provide reconfiguration commands if needed. The server components are listed below.

A. Evaluator: Evaluate the context-aware information and search for appropriate component that is stored in active repository.

B. Component Storage: stores components into the repository.

C. Component locator: Responsible for locating the components in the distributed repository.

D. Optimizer: Implements heuristic algorithms to solve the best configuration problem

E. Client Interface: Implements the communication between the clients and the server.

The repository applies Fuzzy logic algorithm to evaluate the adequacy level of the components and GRASP algorithm to mount the new system architecture.

Figure 6 shows the component diagram of the client module and figure 7 shows the component diagram of the server module.

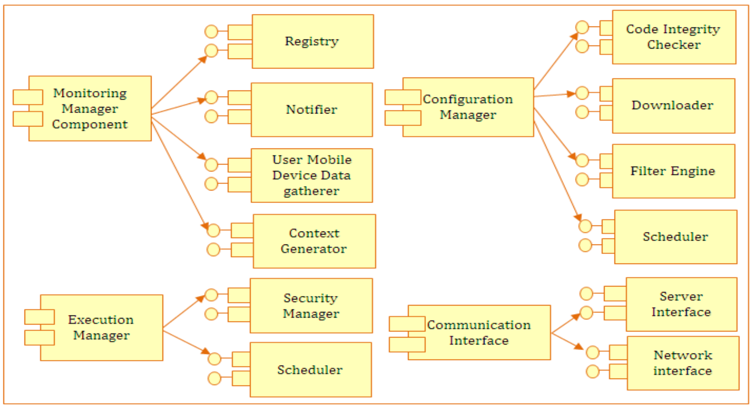

Figure 6. Component Diagram of the Client Module

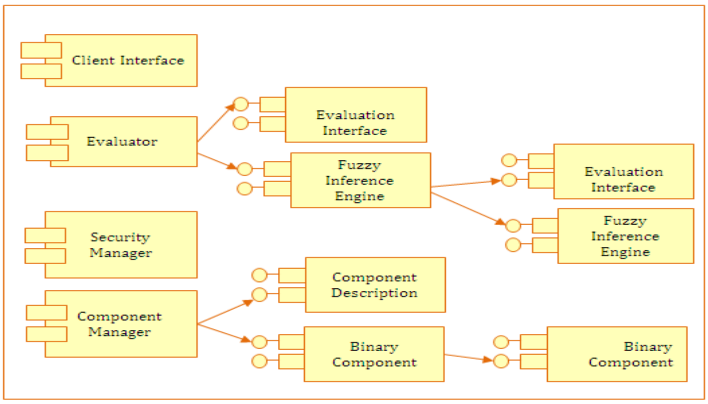

Figure 7. Component Diagram of the Server Module

Figure 8 is an example of class diagram of the proposed context-aware framework for u-healthcare application.

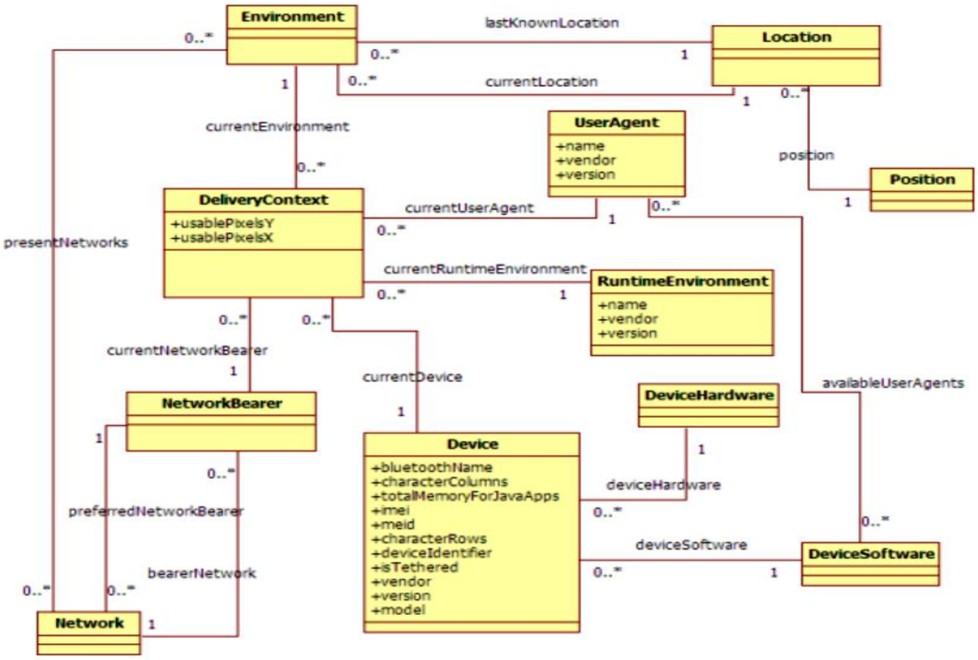

Figure 8. Class Diagram of the Proposed Context-Aware Framework for U-Healthcare Application

4.2.2. Delivering Healthcare Services

The context is generated from the patient environment and sends to the mobile device. The mobile device process the receive data as context. These contexts information will then be process and send to the server. The server read the data and search for the appropriate component from the repository. For example, the sensor send the signal of vital signs of the patient and detected to be in trouble, then the context aware information will be send from user's mobile device to the nearest hospital or call the emergency number to rescue the patient. This context-aware information on the process is raw data which needs to be process and transform into information that the person can understand. Therefore to efficiently process these data, the designed framework which is an active repository for running system will be used. By using the framework, the information gathered from the environment and the patient condition with the help of sensor devices, will be process using the designed component based development and send to the user or the hospital as organized information. This information can be in form of the following:

a. Alert Message Services

1. Low Danger
2. Medium Danger
3. High Danger

b. Location Information Services

c. Condition of the Patient Services

Figure 9 shows the sequence diagram of the proposed u-healthcare application

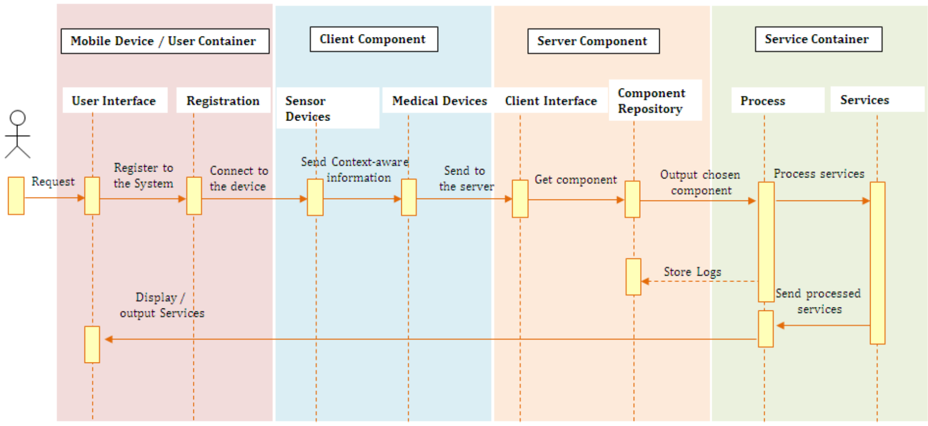

Figure 9. Sequence Diagram of the Proposed U-Healthcare Application

4.2.3. Implementation

The mobile gateway should be hosted on in small and portable device such us smartphone or PDAs. The server side HTTP connects via GPRS, 3G, or 4G enabled mobile internet access. The first and second layer performs the translation between HTTP and Bluetooth communication which run in smartphone in J2ME application and between Bluetooth and Zigbee communication.

The data captured and measured from the wearable monitoring device is further transmitted to the Android smartphone. The Smartphone has developed apps to monitor the data transmitted from the wearable monitoring device. Bluetooth is the connection used to receive the data from the wearable monitoring device. The raw data from the wearable monitoring device and context information are encoded to Android apps. This will be process and real time monitored in mobile device. There are two main functionality of this system. First, the transmitted raw data from wearable monitoring device is send to the clinical specialist that may require a precise description of the recorded information. This can also be needed in the future diagnosis of the patient health status. Second, the presentation of the health status to the users, there are health status indicator for each vital data (Heart rate, Body temperature). From these data, the user may react for example by following the clinical guidelines or recommendations. The app is automatically connected to the healthcare provider. There is another feature that if the health status is determined to be fatal, the app will automatically notify the patient and the hospital for emergency. This information could be (1) alert message, (b) location information, and (c) condition of the patient.

5. Conclusion

In this paper we proposed a context- aware framework for u-healthcare system based on component based - development. This framework process the context from surroundings. We also present the sensor interface framework and multi-purpose gateway to process the contextual data and send to medical centers, hospital or patients mobile device as u-healthcare services. By using the framework, the information gathered from the environment and patient condition will be process using the designed component development and send to the user or hospital in form of different services (1) alert message service, (2) location information services (3) condition of patient services.

Acknowledgement

This research was Supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research

Center) support program (IITP-2015-H8601-15-1007) supervised by the IITP (Institute for Information & communication Technology Promotion).

This research was also supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant number: K 2014075112).

References

[1] J.- T. Park, J. –W. Nah, S. W. Kim, S. M. Chun, S. Wang and S. H. Seo, "Context-Aware Handover with Power Efficiency for u-Healthcare Service in WLAN" 2009 International Conference on New Trends in Information and Service Science.

[2] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research", Dartmouth Computer Science Technical Report TR2000-381.

[3] J. –W. Kim, "A Context Awareness Model for u-Healthcare based on Artificial Neural Network", International Journal of Advancements in Computing Technology(IJACT), vol. 5, no. 12, **(2013)** August.

[4] S. M. Filho, J. Melo, L. E. Leite and G. Lemos, "Context-Aware Adaptation of Component-Based Systems: An Active Repository Approach", Journal of Computer Networks and Communications, vol. 2012, Article ID 963728, pp. 11.

[5] K. Jyrkka, O. Silven, O. Ali-Yrkko, R. Heidari and H. Berg, "Component-based development of DSP software for mobile communication terminals", Microprocessors and Microsystems vol. 26, **(2002)**, pp. 463–474.

[6] P. Coppola, V. D. Mea, L. D. Gaspero, S. Mizzaro, I. Scagnetto, A. Selva, L. Vassena and P. Z. Riziò, "MoBe A Framework for Context-Aware Mobile Applications", http://www.mobe.it/

[7] P. Roocks, M. Endres, A. Huhn, and W. Kießling, "Design and Implementation of a Framework for Context-Aware Preference Queries", Journal of Computing Science and Engineering, vol. 6, no. 4, **(2012)** December, pp. 243-256.

[8] I. Nishimura and S. Xinger, "A Context-aware Song Generator", UbiComp'13, **(2013)** September 8–12, Zurich, Switzerland.