Middleware for Ubiquitous Healthcare Information System

Mangal Sain*, Hoon-Jae Lee** and Wan-Young Chung***

*Graduate School of Design and IT, Dongseo University Busan

**School of Computer and Information Network Engineering, Dongseo Univrsity

***Division of Electronics, Computer and Telecommunication Enginnering, Pukyong National University

Email: wychung@pknu.ac.kr

ABSTRACT

We build middleware architecture with J2EE and LiveGraph to process different ubiquitous healthcare application's data and process that data into useful information, which can play a most important role in decision making in ubiquitous Healthcare System. Application developers mostly rely on third party middleware, tools and libraries (i.e., webservers, distributed middleware such as CORBA, etc.) to respond the emerging trends of their target domain. With this middleware we tried to enhance the efficiency of application by decrease their memory uses, data processing and decision making on another web module which is independent of each application. For middleware system, we proposed an algorithm by which we can find some important conclusion about different health status likewise ECG, Accelerometer. etc., which can be used in various data processing and determine the current health status. In this paper we also analyze some different low level and high level middleware technology which were used to build different kind middleware likewise CAMUS, MiLAN and try to find the best solution in the form of middleware for Ubiquitous Healthcare Information System.

Keyword

Middleware; Ubiquitous; web server, Information

I . Introduction

Healthcare Information system requirements differ from other existing applications. The gap between existing application and medical care needs makes it more complex. To get the maximum benefit of Ubiquitous healthcare system it is necessary to build a Middleware between different ubiquitous applications which can play an important role in management of data and information processing. To enable medical personal for real-time triage, correlation with hospital records and long-term observation of patients is necessary, while sharing the most accurate and real-time patient information. The technology requirements of various domains are increasing, such as distributed platforms are commonly in use, the quality of service requirements for ubiquitous Healthcare systems are increasing as the low level sensor network grows powerful, customers ask for platform independent software and interoperability, etc. Middleware is a class of software technologies designed to help manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application program that provides a common programming abstraction across a distributed system. Middleware can be classifying into two levels, low level middleware and high level middleware. Low level middleware exist between different sensor node, raw data and their application. High level middleware exist between different application which is used for exchange data and communication between various applications. If we first take the consideration then we find there is great work has been done on low level middleware and their different technology. Like Mobiware[1-4], ALICE[5], CAMUS[2], MiLAN[3] etc. but if we talk about high level middleware then we find there is not good work has been done. And we find that without the proper research of

this field we can not achieve the real ubiquitous healthcare system. In this paper, we present a review of the low level and high level middleware literature and then enumerate these common features and then we briefly presents our own middleware system and how it relates to these features.

II. Background

The existing work on middleware. Nowadays, middleware are developed only through industry standards with different variations like J2EE, COM, and CORBA etc. There have been a lot of work done on low level middleware such as Mobiware and ALICE based on CORBA architecture, LIME[6]and ISPACES[7], based on tuple space. These middleware concentrate on communication architecture for ubiquitous system. The existing middleware's for context-awareness are Gaia, Context-Toolkit, CAMUS and RCSM. Communications of these middleware are done using COM, CORBA, java RMI etc. the middleware such as confab focus on privacy sensitivity. All above middleware are based on Low level middleware. The basic reason of negligence of high level middleware because of huge architecture, expensive, time consuming and tones of work load.

The IBM, Red Hat, and Oracle Corporation are major vendors providing middleware software. Vendors such as SAP, TIBCO, Mercator Software, Crossflo, Vitria and webMethods were specifically founded to provide Web-oriented middleware tools. Groups such as the Apache Software Foundation and the ObjectWeb Consortium encourage the development of open source middleware[3].

III. System Design

In this paper we tried to build a Middleware between different ubiquitous healthcare applications so we can process their data into useful information.

Our design prototype consists of a several Healthcare Application (deployed on multiple physical hosts) and a web base application deployed on each ubiquitous application and server which is accessible everywhere through internet. There are several user of that application like Physician, administrator, Patient etc. The main web application is able to collect the whole data onto main server, process them and convert into useful information with the help of LiveGraph[1]. The main controller in this middleware is LiveGraph, which is a centralized service in our architecture. The basic task of the LiveGraph which is stored on application server is to process whole information and set of feedback according to predefined instruction.

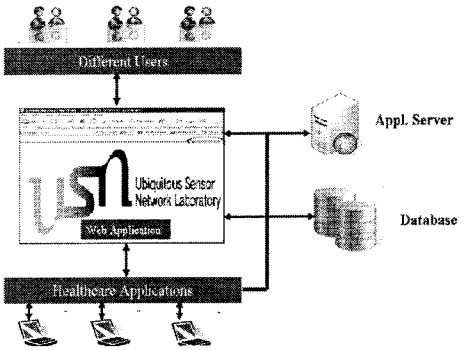

Figure 1. System Architecture.

For example in as per figure 1, if some patient uses our Ubiquitous healthcare system then as per patient register to database his whole data/information will be transferred and analyzed by LiveGraph and stored to database with some specified prescription and feedback. Whole this process has been done without the help of physician because LiveGraph does the whole operation with the help of some specified algorithm which we added in the time of loading the patient data. That information is processed and executed on the application server on LiveGraph. This whole Middleware is distributed into two phase. First one is web application which used as a front end for all user and administrator to access all the information anywhere through internet. Another form is backed where our actual processing has been performed with LiveGraph on application Server. Ubiquitous Healthcare system consist several research levels. This paper mainly focuses on middleware for high level the ubiquitous middleware we focus consists of a distributed environment for servers and various healthcare application.

IV. Frame Structure

We design a high level middleware, that's based on client-server based architecture. As

per figure-1, we can see how client send his request to server with the help of middleware and how both use a common architecture to communicate. To build a perfect middleware we have used Sun Microsystems' Java-based Architecture. We analyzed several architecture such as CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model) etc. We can see in figure-3 client side Applets that can execute their own.

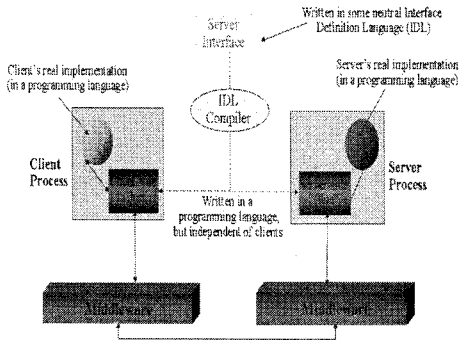

Figure 2. High Level Middleware.

The JVM and Server side web components are used in Servlets and Java Server Page (JSP). Servlets are Java programming language classes that dynamically process requests and construct responses. JSP pages are text-based documents that execute as servlets but allow a more natural approach to creating static content. Although servlets and JSP pages can be used interchangeably, each has its own strengths.

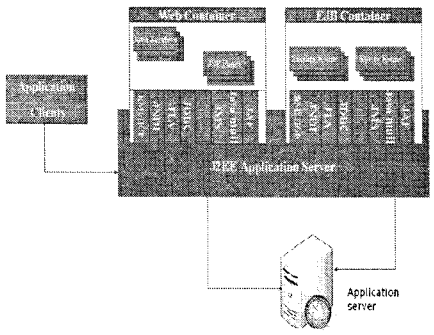

Figure 3. Client-Server Architecture.

Servlets are best suited for service-oriented applications (web service endpoints are implemented as servlets) and the control functions of a presentation-oriented application, such as dispatching requests and handling non textual data. JSP pages are more appropriate for generating text-based markup such as HTML, Scalable Vector Graphics (SVG), Wireless Markup Language (WML), and XML. Enterprises Java Beans (EJB) contains two types of objects such as Session beans and Entity beans. The reason to used J2EE is the power of Java's intrinsic portability graphical, web and garbage collection capabilities. Besides that, J2EE can interface to multiple existing back-end systems and technologies.

V. Simulation Work & Results

To analyze the different application data we take the source code of LiveGraph and modified that into our requirement. As we know WebLogic server is compatible with Java1.5 so we also modified their internal architecture so it can work on WebLogic too. The whole processing is done on LiveGraph and we separate it from web application so we can make faster our web application. The maximum work is done on application server like information processing, analyzing the data etc.

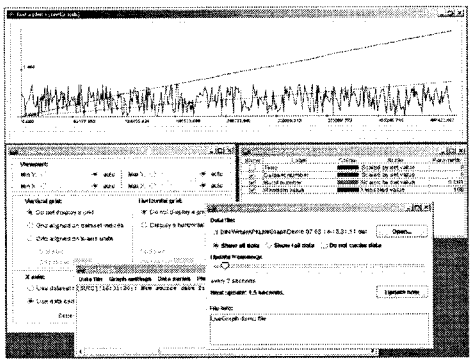

Figure 4. Framework of LiveGraph

LiveGraph is being developed as an exploratory data analysis and visualization framework that allows to visualize the results of application instantly and while it is still running and manage many graphs efficiently. Thus, the frame work for real-time data visualization, analysis and logging.

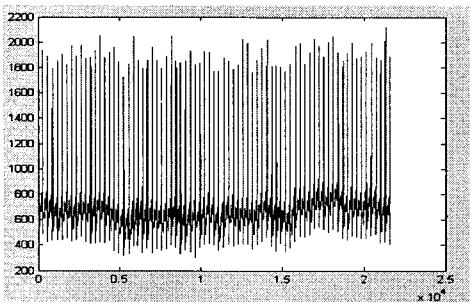

Figure 5. Data analysis using Matlab

Basically we build this whole middleware to minimize the interference of physician. If we analyze the data on Matlab then it need lot of prior knowledge and training but in our application anyone who have some knowledge to operate a simple application can easily operate our application named LiveGraph. Like Patient, Nurse and administrator all can easily operate the web application and specified user can easily operate the whole application because of it is user friendly.

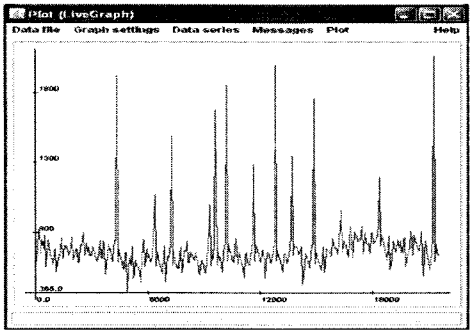

Figure 6. Data analysis using LiveGraph

We can analyze the same data in same way in figure 5 and 6 but we can't utilize the Matlab. We can modify and enhance according to our requirement. With the help of LiveGraph we create middleware which have maximum utilities of Matlab with real time system.

VI. Conclusion & Future Work

There many research in the field of Ubiquitous computing system specially in middleware. In this paper we mainly focus on design of middleware for healthcare analysis application. With the help of LiveGraph and J2EE technology, we build a web application to have an interaction with application server and database. For ubiquitous healthcare information system, we build high level middleware. We implemented in this paper which is just a beginning to build with advance and fully self controlled ubiquitous healthcare system. There are several algorithms which can implement with LiveGraph like measurement of ECG, Accelerometer data, SPO2 etc. In future work, we concentrate on data analysis and implement algorithms for best ubiquitous healthcare information system.

Reference

[1] Live graph. Copyright © 2007 G. Paperin. http://www.live-graph.org

[2] Aekyung Moon, Hyoungsun Kim, Hyun Kim, and Soowon Lee, "Context-Aware Active Services in Ubiquitous Computing Environments," ETRI Journal, Volume 29, Number 2, April 2007.

[3] Heinzelman, W.B. Murphy, A.L. Carvalho, H.S. Perillo, M.A, "Middleware to support sensor network applications", IEEE Network," January/February 2004.

[4] Angin, O.; Campbell, A.T.; Kounavis, M.E.; Liao, R.R.-F.; "The mobiware toolkit: programmable support for adaptive mobile networking," IEEE Personal Communications, August 1998.

[5] Haahr M., Cunningham R. and Cahill V., "Supporting CORBA Applications in a Mobile Environment," Proc. Fifth ACM/IEEE Int'l Conf. Mobile Computing and Networking, MobiCom, 1999.

[6] CampbellA.T., Kounavis M.E., LiaoRRF., and Angin,O, "The Mobiware Toolkit: Programmable Support for Adaptive Mobile Networking," IEEE Personal Communications, 1998.

[7] Murphy, A., Picco,G., and Roman, G.-C. LIME: A Middleware for Physical and Logical Mobility. Proc.21 t International Conference on Distributed Computing Systems ICDCS, 2001.

[8] TSpace Project. 2000. IBM Research. http://www.almaden.ibm.com/cs/TSpaces/.