Communication Middleware For Hospital Automation: Send Alerts and Monitoring of Vital Signs

Cicília R. M. Leite[12], Bruno G. De Araújo [1], Ricardo A. M. Valentim[1], Gláucio B. Brandão[1] and Ana M. G. Guerreiro[1]

[1]Department of Computer and Automation Engineering, Federal University of Rio Grande do Norte – 59075-750 - Natal – RN - Brazil

[2] State University of Rio Grande do Norte and College of Science and Technology Mater Christi - Departament of Computing – State University of Rio Grande do Norte (UERN) 59.625-620 – Mossoró – RN – Brazil

{ciciliamaia,brunogomes, valetim, glaucio,anamaria}@dca.ufrn.br

Abstract. *The development of middleware has emerged as an area of research. It is expanding and is focused the integration of services available for distributed applications. In this context, many challenges have also arisen with the use of middleware, such as communication, flexibility, performance, integration with the Web and the computer itself. The development of middleware for mobile computing incorporates new challenges to developers because of the limitations of the mobile devices. Thus, they have to understanding the new technologies of mobile computing and middleware, in order to integrate them. The objective of this work is to develop a context-aware middleware and service-oriented for data management in real-time using mobile devices.*

1. Introduction

The remarkable advances in wireless communication and distributed systems, and the popularization of mobile devices have turned mobile computing into a reality. Mobile computing is exemplified by devices having some form of processing, such as laptops, PDAs, cell phones, and smart sensors, among others. These devices are related if they are connected through a wired network.

Thus, one can see that the link between the ubiquitous devices favors, which can be understood as the ability to participate in distributed computing regardless of where the device is located. The differences in network connectivity, capacity and platform resource availability can significantly affect application performance. Traditional middleware systems such as CORBA and DCOM have been successful in dealing with heterogeneity in hardware and software platforms that enable portability, thereby facilitating the development of distributed applications.

In this context, we offer appropriate support to address the dynamic aspects of mobile systems. Modern distributed applications require middleware that is able to adapt to environmental changes and is compatible with the level of service quality needed. Using Java technology we developed middleware that will provide data monitoring services, such as real time database access, and send alerts to mobile devices. Thus, the methodology consists of two phases: theoretical and experimental. The theoretical phase consists of reading scientific articles to understand the system, whereas the experimental phase involves system development.

This article is organized as follows: In Section II, we describe the related work. In Section III, we present an overview of the proposed middleware. In Section IV, we present middleware development, its design concept, its implementation, a case study to validate the specific middleware, as well as experiments and results. And finally, in Section V, concluding remarks and a discussion are given regarding future research.

2. Related Work

Continuous data stream processing, emerging as a new research field, concerns the processing of information from sources that produce data at a fast and continuous rate.

For example, information from sensory devices can be considered as a continuously expanding and unlimited sequence of data items without any boundaries. Traditionally, such information required special monitoring applications and equipment that processed and reacted to continual inputs from several sources, such as in a weather monitoring station, patient monitoring equipment, etc.

Advances in electronics have contributed to an increased demand in distributed applications that allow the use of devices with embedded processing power [4]. Examples are industrial networks that use intelligent node controlling processes. According to Pedreiras [12], this occurs because of the trend towards decentralized computing, currently converging to a distributed environment. The functionalities of many processing elements will be processed and compared to centralized computing, which encapsulates functionalities in a single processor with more processing power.

Several studies have been conducted to address various problems faced by industrial automation systems that can potentially be adapted and used in hospital automation. In 1976 Nitzan and Rosen [11] foresaw that industrial automation concepts could be automated using programmable systems such as data acquisition for process control; signal monitoring and processing, providing cost reduction and process optimization. These concepts have been incorporated into the medical environment, making them feasible for use in hospital automation [2].

3. Overview of Middleware

This article presents the development of service oriented context-aware middleware for real-time data management using mobile devices. Middleware tracks data in real time, providing the following services: real-time access to databases from the mobile device; constant database updating from continuous reading data acquisition devices; automatic integration of the various Database Management Systems (DBMS), regardless of their platforms; and the sending of alerts to mobile devices.

After specification and analysis, a prototype will be developed and integrated with a hospital system to detect risk-patients in order to trigger the timely intervention of a Medical Emergency Team (MET).

Hospital systems, highly dependent on information management, must reduce costs and increase efficiency and agility, in addition to ensuring the readiness and effectiveness of the procedures performed. Thus, this project will be of great importance for public health, contributing directly to enhancing services and medical diagnoses, thereby improving the quality of products and services provided to the population.

3.1 System Framework

Capable robust systems are needed to effectively control large volumes of information. Accordingly, middleware was designed with functionality for monitoring information through real-time database queries, in addition to being able to send alerts in critical situations defined by the user. The middleware is based on five stages, as shown in Figure 1.

The first stage is acquisition, responsible for obtaining the data and sending them to a local server, which will in turn relay them to a central server; the second stage involves pre-processing the recovered data, which will be processed by a Digital Signal Processing (DSP) technique such as wavelet transform. In the classification stage, the pre-processed data will be transformed to classify them according to the problem; post-processing is where the data is formatted to be compatible with the system device and are prepared and sent to the registered user of this device. In this stage the middleware is capable of monitoring these data and sending alerts to pre-defined devices.

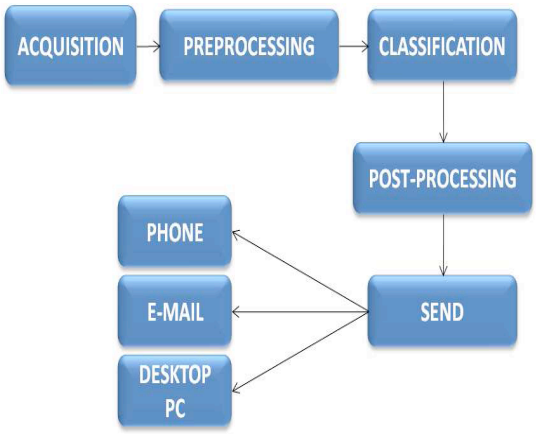

Figure 1. System Framework

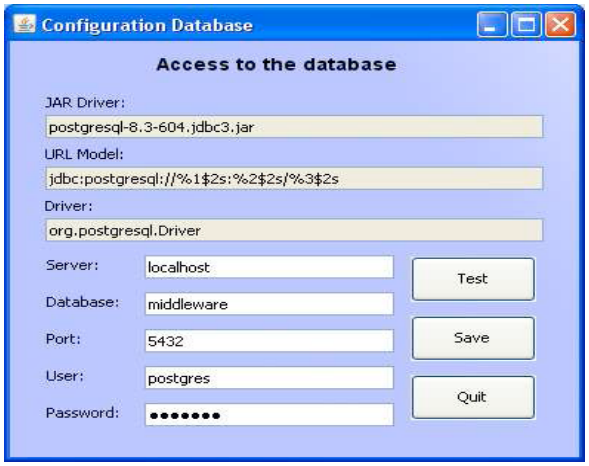

Figure 2. Automatic Configuration Service of the Database

4. Middleware Development

The system offers access to different databases, and the idea is to automate the processes, providing automatic database configuration. The user has only to transmit database information (server, base, port, driver and url model), to be connected automatically, as shown in Figure 2.

Finally, there is a service that sends alerts when critical information is detected. It is able to send Short Message Service (SMS) and alert messages to mobile phones and e-mail addresses, with a predefined message from of a registered user.

To send an SMS, the modem (port, speed, model and brand) must be configured, as shown in Figure 3. To send one by e-mail, server configuration (SMTP, e-mail, user and password) must be performed, as illustrated in Figure 4.

The middleware was developed in Java using the 1.6.0-14 JDK standard library and Eclipse 3.4 environment (Eclipse, 2009). A number of API's and frameworks were also used. The real-time database access was developed with temporal characteristics of the real-time language (LC-RTDB), specified in Leite, 2009 [8].

An automatic database configuration uses JPA, as shown in Figure 4. Configuration of the E-mail technology (Java Persistence API) and sends alerts by SMSLib Framework, which requires the computer to be connected to a GSM modem to send the SMS. An internet connection is the only requirement to send an e-mail.

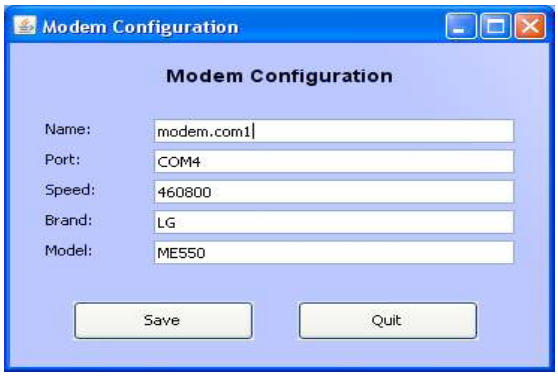

Figure 3. Configuration of the Modem

Figure 4. Configuration of the E-mail

The same middleware specification was used for the development application. The only difference is that the application contains no implemented business logic; it accesses the middleware methods by using its own services.

The middleware developed was applied to the Intensive Care Unit (ICU), where patients require continuous monitoring of vital signs. The application aimed at early detection of risk situations, allowing health professional intervention. This middleware emphasizes the fast transmission to the medical staff of information regarding a patient's vital signs, because it allows prediagnosis when a possible risk is detected. To achieve efficient control, a secure and robust application is required, as illustrated in Figure 5. The middleware developed can be used for this purpose because of the services it offers. The first stage involves registering information on physicians and all the possible ways of sending them alerts, such as email, phone number, etc. This represents the test of middleware services through application.

After registration, an alert will be sent to the doctor by e-mail or SMS if any patient condition falls outside normal limits. A periodic query is set if the patient goes to the ICU, as illustrated in Figure 6. The monitoring screen is best viewed in Figure 7, where the patients conditions are displayed.

$$T_G = \frac{RTT}{2} \qquad (1)$$

In conclusion, the main idea of the middleware tests was to satisfactorily consolidate the functions developed, and validate the services offered through a case study. The middleware can be used in different situations, since domestic monitoring for critical applications shows promising potential.

To measure the performance and efficiency of the experiment were tested by sending several alert messages between a server and a mobile device and vice versa. We used a computer with the following configuration Intel Core 2 Duo 2.4 Ghz, 4GB RAM, Network Card Wireless Intel. The device was a mobile phone with wifi access. Two tests were conducted, each on a different network.

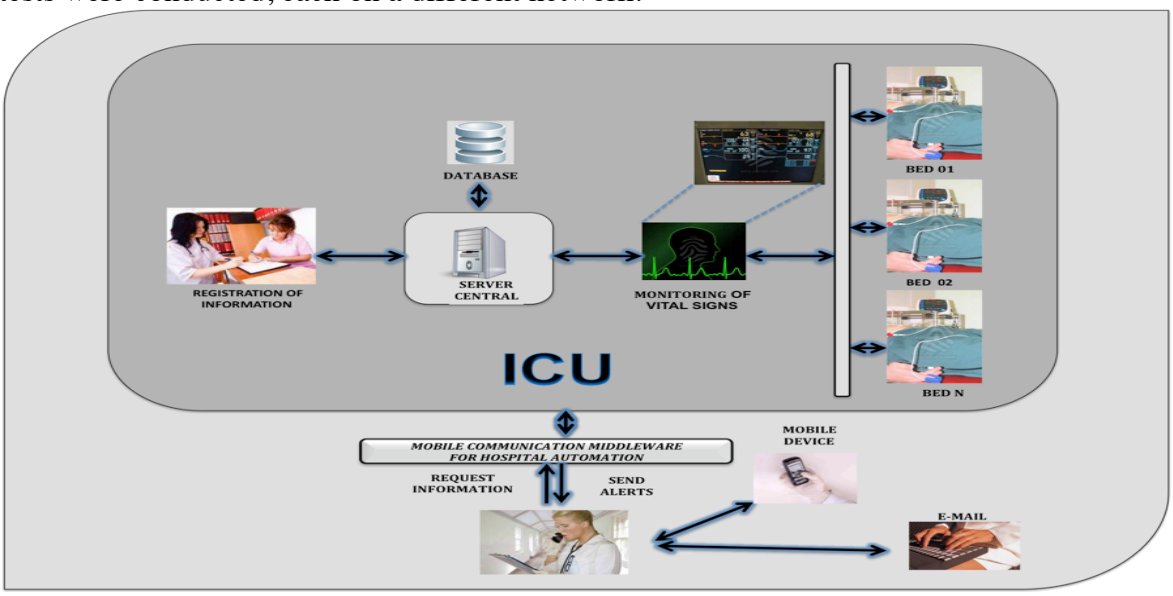

Figure 5. Application the Mobile Communication Middleware for Hospital Automation

To get an estimate of the total broadcasting time occurred during the message transmission system, we use the RTT (Round Trip Time), which corresponds to the time spent and return a message through the network. Thus, using equation (1) can perform this calculation.

Thus, the time spent (Tg) of sending only corresponds to half the RTT. In the test, several items were made and receipts of alert messages to take an average time spent. With the outcome of this experiment, the order to have a idea of the feasibility of using this system in a network environment. The first test was conducted in a residential local area network using a router, 54 Mbps, with the server connected to the router via cable. The second test was performed using a 150 Mbps network with several computers and traffic, in which the server is connected via wireless network. In both tests the mobile device to connect through the wireless network. The result is shown in Table 1

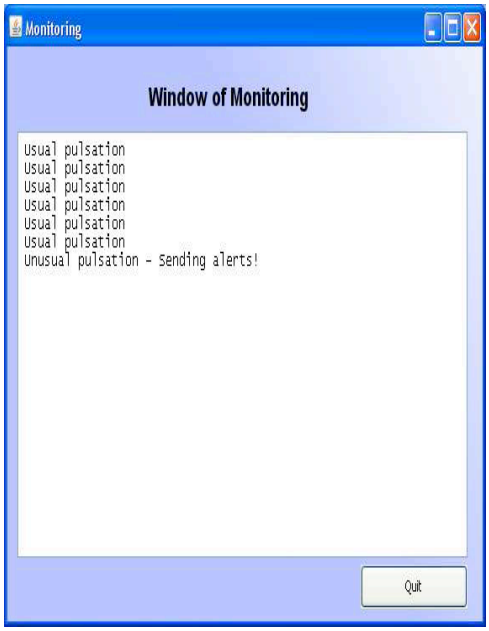

Figure 6. Window of Monitoring

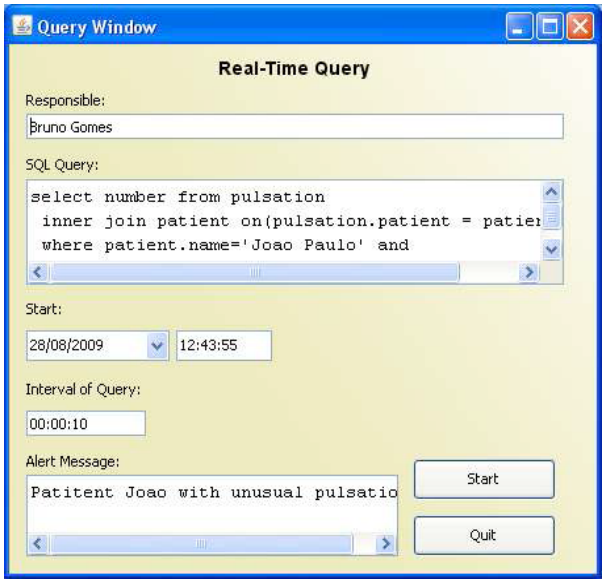

Figure 7. Real-Time Periodic Query

Table 1. Analysis of the peformance - Tests

Network	TEST 01 **(Milisegundos)**	TEST 02 **(Milisegundos)**
1	2	16
2	4	14

5. Conclusions

In this article, we sought to put forth the main concepts of middleware and mobile computing and the mobile computing challenges facing middleware. We developed, a service-oriented context-aware middleware for real-time data management using mobile devices. A case study monitored medical diagnoses by sending the results presented to validate the middleware, services and functions developed. The middleware was used in a hospital setting.

This work, once implemented, will help meet the demand for innovative software and hardware required in automation systems, more specifically in hospital automation. The results of this research project may contribute significantly to information and communication systems in hospitals. Hospital systems are too costly for public institutions to acquire, so our alternative was to develop an open system that enables hospital procedure management. This project will have an important impact on scientific and hospital automation, because it allows the development of an intelligent

system that will help not only doctors but also the entire health team to monitor and diagnose their patients.

Acknowledgment

This work relied on the support of the Laboratory of Hospital Automation and Bioengineering (LAHB) of the Department of Automation and Computer Engineering of the Federal University of Rio Grande of the Norte.

References

[1]Brito, A. E. M., Brasileiro, F. V., Leite C. E., Buriti, A. C. Ethernet Communication in Real-Time to a Netowork of Microcontrollers, Annals of XV Brazilian Conference on Automatica (CBA 2004)-Brazil, September 2004.

[2]Brooks, J.; Brooks, L. Automation in the medical field. Engineering in Medicine and Biology Magazine, IEEE Volume 17, Issue 4, July-Aug. 1998 Page(s):76, 81.

[3]Carreiro, F., Moraes, R., Fonseca, J. A e Vasques, F. Real-Time Communication in Unconstrained Shared Ethernet Networks: The Virtual Token-Passing Approach, submitted at Emerging Technologies and Factory Automation - ETFA, Catania, Italy, 2005.

[4]Dietrich, D., Sauter, T. Evolution Potentials for Fieldbus Systems. WFCS 2000, IEEE Workshop on Factory Communication Systems. Porto, Portugal, September 2000.

[5]Dolejs, O., Smolik, P., e Hanzalek Z. On the Ethernet use for real-time publish-subscribe based applications. In 5th IEEE International Workshop on Factory Communication Systems, Vienna, Austria, Sep. 2004.

[6]IEEE 802.3/ISO 8802-3 - Information processing systems - Local area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2nd edition, 21 September 1990.

[7]Gullikson, M. L. Risk Factors, Safety, and Management of Medical Equipmet. IEEE The Biomedical Engineering Handbook. ISBN 0-8493- 8346-3. USA, 1995.

[8]Leite, C. R. M. Linguagem de Consulta para Aplicacõoes em Tempo-Real [dissertacão], Campina Grande: Universidade Federal de Campina Grande, 2005.

[9]Moraes, R.; Vasques, F., Real-time traffic separation in shared Ethernet networks: simulation analysis of the h-BEB collision resolution algorithm, Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on, vol., no.pp. 89- 92, 17-19 Aug. 2005.

[10] Murakami, Alexandre; Gutierrez,M. A.; Lage, Silvia Helena Gelas; Rebelo, Marina de Fátima de Sá; Ramires, José Antonio Franchini. A Continuous Glucose Monitoring System in Critical. IEEE Computers in Cardiology, v. 32, p. 10-14, 2006.

[11] Nitzan, D.; Rosen, C.A. Programmable Industrial Automation. Transactions on Computers. Volume C-25, Issue 12, Dec. 1976 Page(s):1259 - 1270.

[12] Pedreiras, P., Almeida, L., Gai, P. and Giorgio, B. FTTEthernet: A Flexible Real-Time communication Protocol That Supports Dynamic QoS Management on Ethernet-Based Systems. IEEE Transactions on Industrial Informatics, Vol. 1, No. 3, August 2005.

[13] RFC 4541. Christensen, M., Thrane &Thran, Kimball, K. IGMP and MLD Snooping Switches Considerations. Hewlett-Packard, May, 2006.

[14] Thomesse, J.-P. Fieldbus and interoperability Contr. Eng. Pract., vol. 7, no. 1, pp. 81-94, 1999.

[15] Valentim, R. A. M.; Morais, A. H. F.; Brandao, G. B.; Guerreiro, A. M. G. A performance analysis of the Ethernet nets for applications in real-time: IEEE 802.3 and 802.3 1 Q. Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on 13-16 July 2008 Page(s):956 - 961. 2008.

[16] Várady, P., Benyo, Z. and Benyo, B. An open architecture patient monitoring system using standard technologies. IEEE Transactions on Information Technologies in Biomedicine, Vol. 6, No. 1, pp.95?98, 2002.

[17] Varshney, U. Patient monitoring using infrastructure-oriented wireless LANs. International Journal of Electronic Healthcare, Volume 2, Number 2 / 2006, 149-163, 2006.

[18] Viegas, R. Valentim, R. A. M. Texeira, D. G. Guedes, L. A. Performance Measurements of Protocols to Ethernet Real-Time Applications Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on 20-22 Sept. 2006 Page(s):1352 - 1355. 2006.