Utilising Push and Pull Mechanism in Wireless E-Health Environment

Agustinus Borgy Waluyo[1] Raymond Hsieh[2] David Taniar[2]
Wenny Rahayu[3] Bala Srinivasan[1]
[1]*School of Computer Science and Software Engineering, Monash University, Australia*
{Agustinus.Borgy.Waluyo, Bala.Srinivasan}@infotech.monash.edu.au
[2] *School of Business Systems, Monash University, Australia*
David.Taniar@infotech.monash.edu.au
[3] *Department of Computer Science and Computer Engineering, Latrobe University, Australia*
wenny@cs.latrobe.edu.au

Abstract

The emerging of wireless computing motivates radical changes of how information is derived. Our paper concerns with developing push and pull based application in wireless environment. We use a simplified e-health (hospital) context to demonstrate some effective uses of pull-based and push-based mechanisms. The pull mechanism involves information retrieval from a database, and update information in the database. The push mechanism is classified into three scenarios, (i) aperiodic push, (ii) periodic push – global recipients and (iii) periodic push – selective recipients.

1. Introduction

Recent advances in wireless technology have led to mobile computing, a new dimension in data communication and processing. Many predict a new emerging, gigantic market with millions of mobile users carrying small, battery-powered terminal equipped with wireless connection [1,3].

In this paper, we apply pull-based and push-based mechanisms in wireless environment. We use a simplified e-health (hospital) context to develop a hospital server and client application. The hospital context relates to doctors as the principal clients to a server application. The application will demonstrate the usability of wireless networks, and improve the mobility of doctors through wireless data dissemination

There are two ways of data delivery in wireless environment. One is called *pull* mechanism, and the other is *push* mechanism [2] Pull mechanism is when the data are delivered on a demand basis. In the e-health context, we apply this mechanism for doctors to retrieve his/her patients. In push mechanism, the server initiates the delivery of data without a specific request from the client. We apply this mechanism to send a direct message to a specific doctor, and to distribute information to all or selective doctors such as news bulletin.

Push mechanism can be categorized into 1-1 (unicast) and 1-N communication type. Unicast communication involves a server and a client, and the data is sent from the server to the client. 1-N communication can be either multicast or broadcast mode. In multicast mode, the recipients are known and the data are delivered only to those recipients, for example; the information is delivered to doctors and nurses that are registered in a specific domain. On the contrary, the broadcast mode simply sent the data without knowing the number of clients who might receive the data. This paper concerns with 1-N (multicast mode) communication type.

The subsequent sections of this paper are organized as follows. Section 2 describes the proposed model. Section 3 presents the implementation of pull-based and push-based mechanisms in e-health context. Finally, section 4 concludes the paper.

2. Proposed Model

The proposed model comprises of three specific components. The three components are: (1) a data source, (2) a server application, and (3) a client application. The data source component is a simple Microsoft Access® database. The data source and server components are located on the same server device. The server component is an application that functions as a mediator between the data source and client application. The client application accepts push-based information from the server; or the client can create requests for processing (by the server application). It does not access the data source directly. The connection over the TCP/IP network is created via the Winsock controls on both the server and client application. The Winsock control allows for both pull-based and push-based communications.

Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)
0-7695-2073-1/04 $20.00 © 2004 **IEEE**

3. Wireless E-Health System: Implementation

The proposed scenario uses a simplified e-health context to develop a hospital server and client application. The application aims to demonstrate two uses of pull-based and push-based communications. The four techniques for demonstrating the dissemination of data are as follows: (1) a pull-based event to retrieve information from a database, (2) a pull-based event to update information in the database, (3) a push-based event to send a simple message to a client, (4) a push-based event to multicast information to all clients, as well as to multicast information to selective clients.

3.1 Technology

The hardware technology we use is a laptop computer as the server device. Although a laptop computer is a portable device, its primary function is to behave as a server device in regards to the application. The client device is a Pocket PC-based Personal Digital Assistant (PDA). Hewlett Packard iPAQ™ H5450 Pocket PC is the type of PDA that we use. The PDA communicates with the server device over a wireless LAN. Since the transmission medium is wireless, the nominated wireless transmission standard is 802.11b. As for the software technology, we utilise Microsoft® WindowsTM 2000 for operating system software for the server device, and Microsoft® Pocket PC 2002 for the client system software. Furthermore, Microsoft® Pocket PC 2002 also features a set of network-related services required by the proposed model.

Two software development products that we employee are Microsoft® Visual Basic® 6.0 and Microsoft® eMbedded Visual Basic® 3.0.Microsoft® eMbedded Visual Basic® 3.0 is an object-oriented/event-driven high-level programming language. It is similar to Microsoft® Visual Basic® 6.0; except, it is a software component that enables Visual Basic® 6.0 applications to connect to various data sources such as a Microsoft® Access®2002 database to perform search and update functions.

The service device and the Personal Digital Assistant both use 802.11b to create a wireless ad-hoc network [4]. The TCP/IP protocols are used to manage the transmission of messages between the server device and client device over the 802.11b wireless network. TCP/IP is a necessity on the server device and client device because the two software development programs use a control called Winsock [5]. The Winsock control is added to both the server application and client application at design time. This enables the server application and client application to communicate messages over a TCP/IP network.

3.2 Database setup

The database utilized by the application comprises of three related tables. The three tables entitled ***Doctors,*** ***Patients,*** and ***Remote,*** are given in Figure 1.

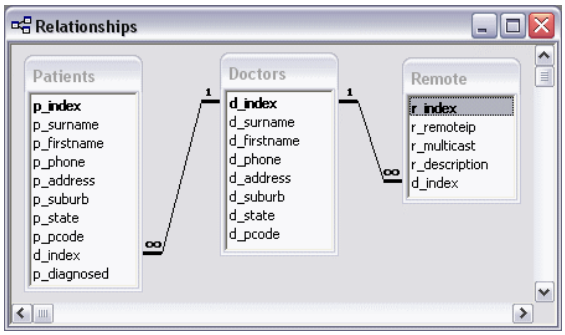

Figure 1. Table relationships for hospital database

The ***Doctors*** table stores records of doctors employed by the hospital. The ***Patients*** table stores records of patient details; including the doctor assigned to care for the patient. Finally, the ***Remote*** table stores a list of IP addresses. Each IP address stored in the ***Remote*** table is assigned to a doctor.

3.3 Pull 1: Client retrieving data from the database

Each time a doctor connects to the server application, the doctor may retrieve a list of patients from the database.

If a doctor wishes to retrieve a list of patients, the doctor simply clicks the '*My Patients*' button on the client application. The image shown in Figure 2 highlights the 'My Patients' button on the client program. Figure 3 illustrates the client application displaying patient details

Figure 2. The 'My Patients' button

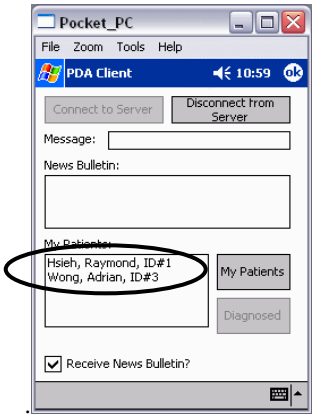

Figure 3. Client details returned for display

3.4 Pull 2: Client updating the database

As an extension to retrieving patient information from the database, the client application also enables the doctor to diagnose a patient.

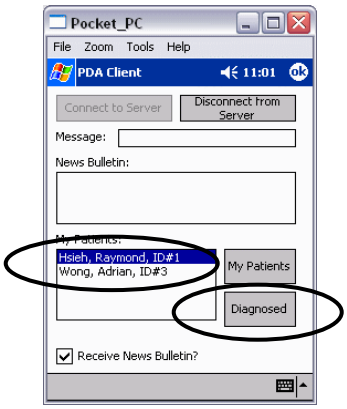

Figure 4. Selected patient and 'Diagnosed' button

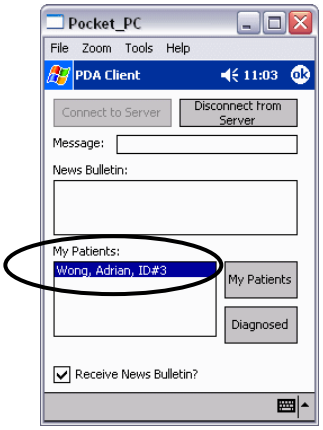

Figure 5. Patient removed from client display

Once a list of patients is retrieved from the server application, the doctor may click the '*Diagnosed*' button to remove a selected patient from the list of patients. The image shown in Figure 4 illustrates a selected patient and the relevant '*Diagnosed*' button. Once a patient is diagnosed, the list of patients on the client application is updated to reflect the changes made to the ***Patients*** table. The image in Figure 5 illustrates the updated client display.

3.5 Push 1: Sending a message from server to client

Whenever a doctor connects to the server, a list of all doctors currently connected is displayed on the server application. This is shown in Figure 6.

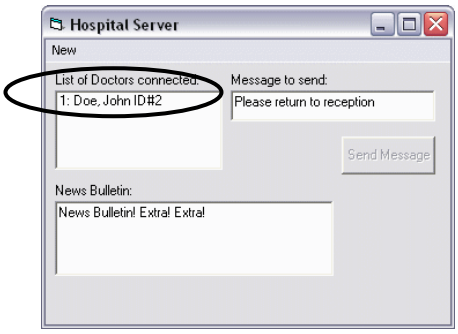

Figure 6. One client connected to the server

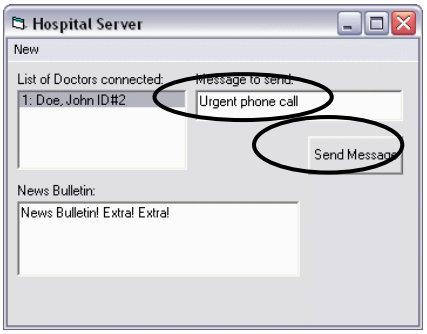

Figure 7. Sending a message from server to client

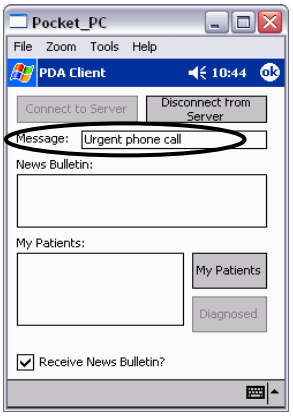

Figure 8. Result of sending message at client side

The list is created or updated whenever a doctor connects or disconnects from the server application.

When a doctor attempts to connect the server, a connection request is sent via Winsock Connection Request. It will then create a connection for each connection request; and adds the details of the doctor to the list. To send a message to the client, the server operator simply: selects the desired client from the list; types the desired message into the '*Message to send*' text box; and clicks '*Send Message*'. This is shown in Figure 7. Whilst, Figure 8 demonstrates the result of the server application sending a message to the client application.

3.6 Push 2: Multicasting information from server to client(s)

The server application automatically sends a news bulletin to each doctor connected to the server. It uses a timer to periodically send the news bulletin to each doctor every twenty-five seconds. Figure 9 shows an image when the server sends a news bulletin to doctors.

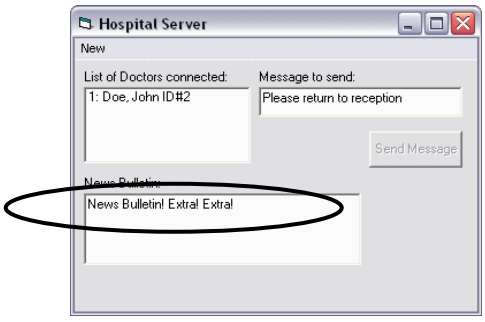

Figure 9. Sending a news bulletin to doctors connected to the server application

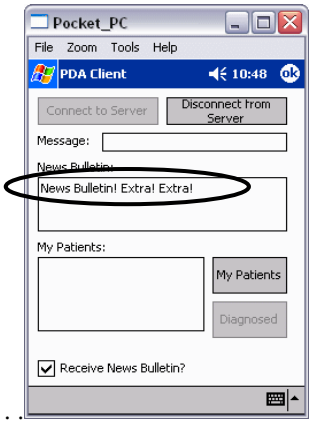

Figure 10. Result of sending the news bulletin

To modify the contents of the news bulletin, the server operator simply changes the text in the '*News Bulletin*' text box. When the twenty-five second interval expires, the new contents of the '*News Bulletin*' text box are sent to each doctor who requires a copy of the news bulletin. The image shown in Figure 10 demonstrates the result of the server application sending a message to the client application.

Client can enable or disable the automatic sending of the news bulletin by selecting or de-selecting the check box marked '*Receive News Bulletin?*". Figure 11 illustrates this feature.

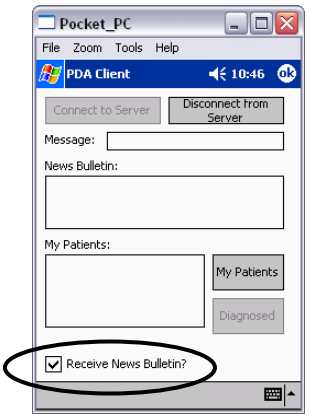

Figure 11. Check box for receiving news bulletin

4. Conclusions and Future work

In this paper, we have presented application using a simplified e-health (hospital) context to demonstrate some effective uses of pull-based and push-based mechanisms. The hospital scenario relates to doctors as the principal clients to a server application. Furthermore, it demonstrates the usability of wireless networks, and to improve the mobility of doctors through wireless data dissemination.

For future work, we plan to incorporate context-awareness ceoncept. This will enhance the flexibility and efficiency of the activity.

5. References

1. Acharya S., Alonso R., Franklin M. and Zdonik S., "Broadcast Disks: Data Management for Asymmetric Communication Environments", *In Proc. of ACM Sigmod*, pp.199-210, 1995.
2. Aksoy D., Altinel M., Bose R., Cetintemel U., Franklin M., Wang J., and Zdonik S., "Research in Data Broadcast and Dissemination", *In Proc. of AMCP*, LNCS, **1554**:194-207, 1999.
3. Barbara D., "Mobile Computing and Databases-A Survey", *IEEE TKDE*, **11**(1): 108-117, 1999.
4. Blake, R., *Electronic Communication Systems*, 2nd Edition, Thomson Learning, Inc., New York, U.S.A, 2002.
5. Zak, D., *Programming with Microsoft Visual Basic 6.0*, Thomson Publishing, U.S.A., 2000.