STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 1, 2007

EFFICIENT DATA SYNCHRONIZATION FOR MOBILE WIRELESS MEDICAL USERS

ADRIAN SERGIU DARABANT AND HOREA TODORAN

Abstract. In order to take the appropriate decisions as quick as possible, medical doctors need fast access to various pieces of information on their patients. The required information should be accurate, up-to-date, and available on the spot. Even more, after finishing his/her investigation, the medical doctor should be able to immediately forward the relevant results to other medical personnel (doctors, nurses, hospital administration). All this could be implemented in a mobile solution using handheld devices. The goal of our paper is to present a pilot implementation of a medical database system with dynamic and efficient data synchronization using wireless technologies.
Key Words: mobile applications, wireless data synchronization, personal digital assistants, medical applications.

1. Introduction

Being able to perform virtually the same tasks as desktop computers or laptops, Personal Digital Assistants (PDAs) are constantly increasing their attractiveness. Although originally created to run simple applications like electronic diaries, address books or planning calendars, the handheld devices eventually evolved into real pocket computers capable of undertaking more complex tasks, from word processing and spreadsheet editing to multimedia authoring. Current models support data transfer over communication networks via the common wireless protocols (infrared, bluetooth, WiFi, GPRS), therefore providing access to convenient services, including web browsing, messaging, email, and so on.

Besides the *lower power consumption* (the battery life is approximately two times longer than in the case of laptops), the most important advantages of the handheld computers over other mobile devices achieving similar performance consist in their higher *portability* and *mobility*. PDAs are much lighter than laptops or TabletPCs (approximately 100-200g), fit into the jacket's pocket (wearable), can be hold into one hand and operated with the other (handhelds), and can be operated even on the move. Furthermore, they are very easy to use and prove

Received by the editors: August 1, 2007.

2000 *Mathematics Subject Classification.* 68P15.

1998 *CR Categories and Descriptors.* D.2.11 [**Software Engineering**]: Software architectures –*Domain-specific architectures*.

economic viability [1], having much of the computing capability and storage capacity of laptops at a fraction of the cost (some authors even called them "equity computers"- e.g. Andrew Trotter in [2]). As a consequence, PDAs are more and more exploited in various fields, including mobile business ([3]), mobile education, medicine, and leisure.

Nonetheless, there are also inconveniences when using handheld devices. The most significant are related to the *small size of the screen*, which confines the amount of information displayed or requires the intensive use of navigation bars. *Data input* is more difficult than in the case of desktop computers or laptops, since the keyboard and the mouse are very small (if present). Even the stylus pens are rather narrow, therefore requiring accurate operation on the screen pad. Handhelds have relatively limited storage capabilities, are difficult to upgrade and much less robust than TabletPCs, laptops, and desktop computers. Taking all these restrictions into account, software producers must design applications running on PDAs more carefully than those for the other types of PCs. As a good practice example, the Windows Mobile for PocketPC family includes scale down versions of the Microsoft Windows operating systems, very similar to the desktop versions, though adapted in terms of minimal requirements (memory, processor speed) and visual elements (windows, menus, lists, buttons) to the characteristics of PocketPCs. An exhaustive list of general design requirements for Windows Mobile-based PocketPC applications is given in [4].

2. Problem Formulation

Medical doctors need to be efficient when consulting their patients in the daily routine, in terms of both saving time and taking the appropriate professional decisions. A crucial prerequisite for achieving a high level of efficiency is the quick access to a whole range of information about the current patient: medical history, results of previous medical investigations, opinions of other specialists on the case, and so on. Moreover, the medical doctor should be able to easily disseminate the results of his/her own investigations on the patient.

The traditional solution is to use paper-based patient files that have to be carried by the medical doctor or by the accompanying staff to the patient's bed. The more information required for a certain investigation, the bulkier the dossier and the more difficult the search for relevant data.

A modern alternative is to use a computer-based solution, with a piece of equipment that is small enough to fit easily into the jacket's pocket, and with sufficient computing and communication power to rapidly bring patient's data on demand from a central database onto the device.

Although there are still situations where it is feasible for medical doctors to carry the paper-based patient files with them, these do not always provide the most up-to-date information, which is essential when taking the decisions. For instance, the most recent results of laboratory investigations might have not yet been recorded in the dossier. Furthermore, even if the raw data has already been recorded, it has not yet been interpreted according to medical procedures.

In this paper, we propose an innovative software architecture on mobile devices with communication capacities, solution that can be easily used with any exiting medical data management software. The application not only facilitates the management of various sets of information (doctors, patients, investigations, and so on), using the PDA, but also synchronizes the data between the mobile device and the hospital's database server.

We implemented the proposed framework on mobile assistants (PDAs) with cellular phone and/or wireless 802.11 capabilities [6, 7] in a pilot project that has been experimentally linked with the management system of a hospital. With the aid of the new framework a hospital that has many premises is able to manage, at any moment, updated information about its patients, even coming from far locations. It is also able to keep in contact with general practitioners who, after visiting their patients at home, can immediately send the results of their investigations to the specialists.

The novelty of our system consists in using an incremental data synchronization mechanism based on timestamps, as described in section 4.1 below. The system architecture (see section 3 for details) also ensures a high degree of independence between the mobile system and the hospital's data management system, which is crucial in case of temporary failure of one of the components. Optimized network traffic is achieved by means of a data compression solution, illustrated in section 4.3 of the present paper.

2.1. **Data flow.** The system relies on the existence of the hospital's central database server to which medical staff (doctors, nurses, laboratory assistants), as well as administrative personnel shall have access anytime and from any location within hospital's premises. They not only frequently ask for updated information concerning the patients, but also send new information to the database server (e.g. results of their investigations).

Medical doctors and their assisting staff query the hospital's database to get useful information that could help them conducting their investigations on the patient. As a result of these investigations, the medical staff attains new information that should be inserted into the database for future use. Various pieces of the recorded information could also be used by the administrative staff, for instance to estimate the cost of the treatment or to plan the use of equipment.The Fig. 1 below depicts the typical information flow within the medical architecture we are trying to model.

The general practitioner is in our scenario a mobile user that travels with his/her PDA. Usually, traveling means either visiting the patients in the hospital, or receiving them in his/her office, but general practitioners might as well visit the patients at their residence in certain situations. Given this *volatile* environment,

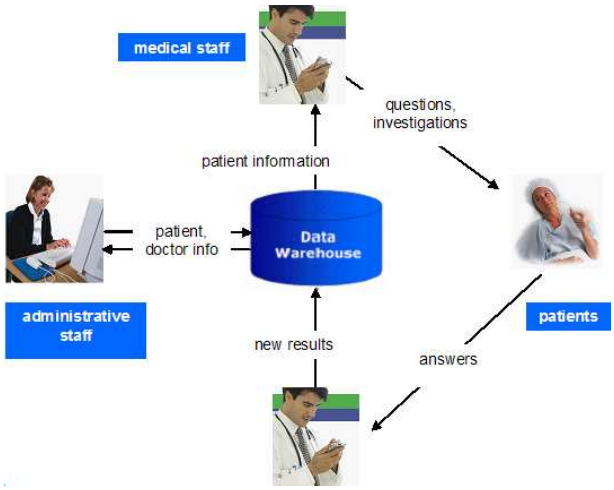

FIGURE 1. Medical information flow

we would like the medical doctor to have as much information as possible about the patients he/she is consulting at a given time. Furthermore, a medical doctor could get assignments for visiting patients directly. The patient's consultation request at the clinic is either directly pushed by the software system on the general practitioner's mobile device or collected in the course of the next data synchronization between the central system and the mobile device.

During a consultation the general practitioner usually evaluates the patient's state using various medical analysis methods: blood sampling, temperature, blood pressure, etc. The results of all these measurements consist in various data sets that the general practitioner would normally write in the patient's record. Using our mobile application - *MobMed* - the general practitioner no longer needs to fill in paper reports and records about the patient. All he has to do is to input the various data sets he obtained into the database, using the friendly interface we designed. Data is temporarily stored in its own portable device (PDA) and then synchronized with the central warehouse of the hospital. Virtually, the general practitioner has a small sized tool capable of storing patient's records, recalling history data about the patients and therefore helping the medical doctor to establish an early potential diagnosis.

3. System Architecture and Services

In the following paragraphs we present the proposed system architecture that has been already implemented in a pilot project called *MobMed*. One of our major requirements was to build a system that is as least intrusive as possible in the existing *hospital* or clinic *management software* (HMS). With this goal in mind, we needed to link the mobile system facilities to an existing software solution in such a way that no major adjustments are required for the already implemented solution.

We used the following model to implement the mobile architecture on the top of an existing hospital management software, which is running on an MS Windows-like operating system and storing data in an SQL Database Server. We present this particular architectural model here in order to show the degree of independence between the mobile system and the existing management solution.

Even if it is a challenging task, the integration between any existing management software (HMS) and *MobMed* is always possible with little or no intrusion into the HMS at all. In order to implement the *MobMed* solution we suppose that the clinic (hospital) already has at list an Intranet system, possibly connected to the Internet by some highly secure connection. We further suppose that, in most of the cases, the internal network of the hospital is protected by a firewall that separates the Intranet from the Internet.

In Fig. 2 below the general practitioners are running MobMed on Pocket PC devices with incorporated GSM phones or wireless radio cards (WiFi). The choice of GPRS/UMTS or wireless is conditioned by the necessary connection persistence. If the connection is desired at any particular moment, the best choice would be GPRS/UMTS, both requiring a GSM line. For connections only in hospital's perimeter the use of the 802.11 standard and WiFi hotspots might be a convenient alternative.

Since we want to give access to the mobile general practitioners to the database server where patient, diagnosis and consultation schedules data are stored, some features of the Microsoft based platforms can be used. First of all, the SQL Database Server, in its 2000 and 2005 variants, offers support for mobile subscribers to data publications.

The *MobMed* server runs SQL Server and provides for data synchronization with the mobile devices. The synchronization procedure uses the HTTP protocol for data transport in order to be easily accessible on highly secured platforms. The mobile devices are running a .NET application platform with Mobile SQL Server 2005 as data storage and synchronization engine. Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile), the "descendant" of Microsoft SQL Server 2000 CE Edition 2.0 (SQL Server 2000), extends the Microsoft enterprising solutions for *line-of-business* and for the management of personal information applied on a device. SQL Server Mobile delivers the functionalities necessary to a relational database, transposed to a lower scale: robust information storage, query preparation, connectivity capacities. Microsoft SQL Server 2005 was projected to support an extended list of mobile devices and Table PCs. The mobile devices include any device that runs Microsoft Windows CE 5.0, Microsoft Mobile Pocket PC 2003,

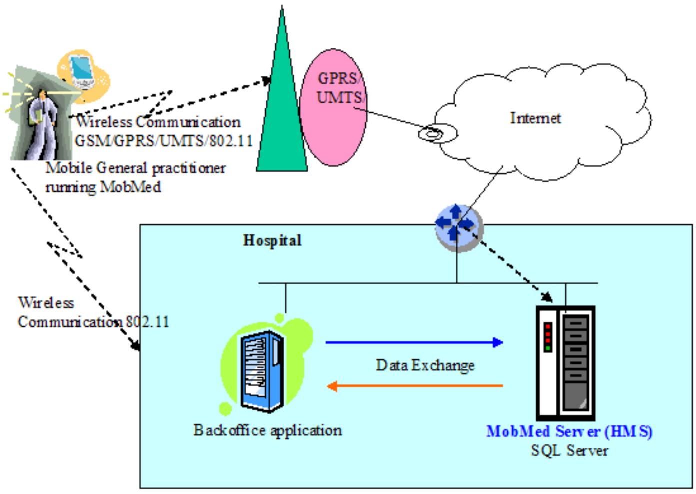

FIGURE 2. The MobMed Architecture and Integration with the Hospital Management Software

Microsoft Mobile Version 5.0 Pocket PC, or Microsoft Mobile Version 5.0 Smart Phone.

Merge replication[1] is an ideal synchronization mechanism when using mobile devices, as it allows the automatic and independent data update on both the mobile device and the server. As soon as the device is connected, the data is synchronized, changes being sent from the client to the server, along with the new information being pushed from the server to the mobile device. Merge Replication needs more server configurations and maintenance than its alternative method (Remote Data Access), but it is more advantageous for applications that involve several mobile devices. Merge Replication also has the capacity of detecting and solving shown up conflicts and of rejoining data from several tables at the same time. It allows instrument survey by using SQL Server and provides more data rejoining options such as the article types and filtering.

Merge Replication is a proper solution when conflict solving is required, when information has to be propelled to and from the desktop or laptop computers and when working with larger databases.

Merge Replication uses some components of the Microsoft SQL Server 2005 Mobile Edition (SQL Server Mobile): SQL Server Mobile Database Engine, SQL Server Mobile Client Agent, SQL Server Mobile Server Agent, SQL Server Mobile Replication Provider.

---

[1]Microsoft TechNet[http://technet.microsoft.com/en-us/library/ms171927.aspx]

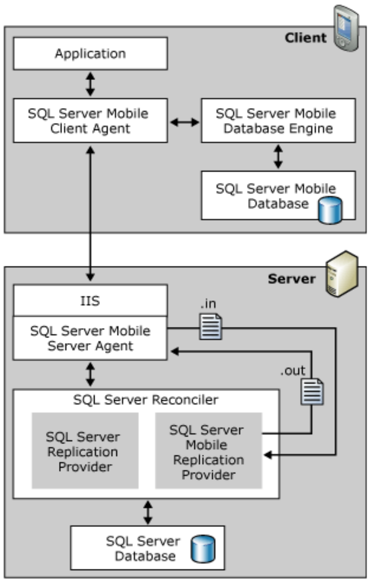

FIGURE 3. Merge Replication Architecture

4. Data Exchange and Synchronization

As already mentioned, the synchronization method uses HTTP as transport protocol. HTTP is used because of its high implementation availability on all platforms and because it allows easier through firewall communication.

4.1. **Merge replication implementation.** The functioning of *merge replication* from Microsoft SQL Server 2005 Mobile Edition implies the following processes:

(1) The data is published on the SQL Server
(2) A *subscriber* is created for the publication
(3) The data in the subscriber is updated
(4) The data is synchronized.

Details on data publication and subscriber registration setup are presented in [8].

When a SQL Server Mobile *subscriber* is synchronized with the SQL Server, all the changes upon data are recovered by the database publication. However, when a SQL Server Mobile subscriber is synchronized for the first time, it can recover the data either directly from the database publication or from the snapshot file. The sealed class *SqlCeReplication* is part of the space names *System.Data.SqlServerCe* and allows the synchronization of SQL Server Mobile databases with the SQL Server databases. It's interface is described in [9].

The typical steps to perform data synchronization when everything is setup are depicted bellow:

```
SqlCeReplication repl = null;
Try{
    repl = new SqlCeReplication();
    repl.InternetUrl = "http://www.myHMSserver.ro/sqlmobile/sqlcesa30.dll";
    repl.InternetLogin = "MyInternetLogin";
    repl.InternetPassword = "<password>";
    repl.Publisher = "MyPublisher";
    repl.PublisherDatabase = "MyPublisherDatabase";
    repl.PublisherLogin = "MyPublisherLogin";
    repl.PublisherPassword = "<password>";
    repl.Publication = "MyPublication";
    repl.Subscriber = "MySubscriber";
    repl.SubscriberConnectionString = "Data Source=MyDatabase.sdf";
    repl.AddSubscription(AddOption.CreateDatabase);
    repl.Synchronize();
} catch (SqlCeException) {     // Error handling/ }
finally   {    repl.Dispose();  }
```

4.2. **Conflict Resolution.** An important aspect in data synchronization with multiple data sources is conflict resolution. Devices might update the same entity locally with different data and then upload changes to the *MobMed* server in a random order. The problem is choosing the correct final value/state of the entity. Automated conflict resolution is usually tightly dependent on the business rules that govern the conflicting entity's type ([10]). SQL Server Mobile detects *client-side* conflicts but does not manipulate their settlement. The conflict information is sent to the Publisher in view of settlement during the following synchronization. Most of the conflicts are settled by the Publisher following the synchronization.

4.3. **Data Compression and Features.** An important aspect when dealing with GSM transports is the cost of the data transfer. As the GPRS/UMTS solutions are still costly, the employment of a data compression mechanism is beneficial. The application has native compression support based on Merge replication compression. Another feature that reduces traffic is column and cell based synchronization. When just a single column of database row has been updated, only that value is sent to the server avoiding thus an entire row transmission.

As already mentioned in the introductory section of this paper, software applications for PDAs have to be carefully designed in order to overcome the limitations of the handheld devices, especially those related to the small size of screen and the difficult use of data input accessories (keyboard, mouse, stylus pen). We certainly took these constraints into account when designing the user interface of the *MobMed* solution.

Following are the main characteristics of the *MobMed* user-friendly interface:

- the navigation bar always displays the name of the topmost window, thus avoiding confusion;
- common menus appear on the leftmost position of the MenuBar in a known order;

- whenever text fields are present on the screen, they are accessible with the Soft Input Panel (SIP) up, thus facilitating data input from mobile devices without keyboard;
- the application maintains the regional and language settings specified by the client.
- in order to ease user's navigation through the successive application's windows, we employed suggestive graphical elements wherever appropriate. It is also the case of the main menu, made of attractive graphical buttons.

5. Conclusions and future work

The current paper proposes a novel multi-tier system to enhance the mobility of medical staff, thus bringing an important efficiency advantage to the hospital/clinic implementing it.

The main functional characteristics of our proposal are related to medical staff having access to up-to-date, complex information on their Pocket PCs, virtually wherever and whenever they need it. This is provided by on demand secure and fast synchronization of the local database from the mobile device with the central data warehouse of the hospital.

The application running on the mobile device (MobMed) has a user-friendly interface, which is designed according to the general requirements described in [4].

As far as the system architecture is concerned, our model ensures application isolation and independence in the case of temporary failure between the three tiers: the mobile device, the data management server, and the client's existing management software. Moreover, *MobMed* successfully deals with important aspects related to data synchronization, like conflict resolution and data compression.

Incremental data synchronization, optimized network traffic, and cvasi-permanent availability are features that make our system valuable and different from other similar implementations.

Future work is intended to building a general framework allowing the smooth integration with any HMS. It should also improve the security of data exchange and the control of mobile agents' navigation from the handheld devices.

References

[1] E. Dieterle, *Handheld devices for ubiquitous learning and analyzing*, 2005 National Educational Computing Conference, Philadelphia, PA, available at [http://center.uoregon.edu/ISTE/uploads/NECC2005/KEY_7287575/Dieterle_NECC2005Dieterle_RP.pdf], visited June 2007.

[2] A. Trotter, *Palm computing moving from the workplace to the classroom*, in Education Week, October 1999.

[3] A.S. Darabant, H. Todoran, *Building an Efficient Architecture for Data Synchronization on Mobile Wireless Agents*, in WSEAS Transactions on Communications, Issue 8, Volume 5, August 2006, ISSN 1109-2742, pp. 1384-1391.

[4] ***, Designed for Windows Mobile *Software Application Handbook for Pocket PCs*, Microsoft Corporation, May 2004.
[5] M.D. Sutton, *Data Synchronization: Which Technology?*, Intel Software Network, [http://www.intel.com/cd/ids/developer/asmo-na/eng/52893.htm], visited June 2007.
[6] ***, *IBM eServer i5 and iSeries System Handbook: IBM i5/OS version 5*, IBM Corporation, October 2004, [http://www.redbooks.ibm.com/redbooks/GA195486/wwhelp/wwhimpl/java/html/wwhelp.htm], visited June 2007.
[7] M. Mallick, *Mobile and Wireless Design Essentials*, Wiley Publishers, 2003.
[8] Microsoft TechNet, emphHow to: Create a Publication and Define Articles (SQL Server Management Studio), Microsoft Corporation, [http://technet.microsoft.com/en-us/library/ms151160.aspx], visited June 2007.
[9] MSDN, *SqlCeReplication Class*, Microsoft Corporation Documentation, [http://msdn2.microsoft.com/en-us/library/system.data.sqlserverce.sqlcereplication.aspx], visited June 2007.
[10] W. Wheeler, *Integrating Wireless Technology in the Enterprise*, First Edition: PDAs, Blackberries, and Mobile Devices, Elsevier Digital Press, 2003.

Babes-Bolyai University, Cluj-Napoca, Romania
E-mail address: dadi@cs.ubbcluj.ro

Babes-Bolyai University, Cluj-Napoca, Romania
E-mail address: dadi@cs.ubbcluj.ro