DISTRIBUTED PREDICTIVE HEALTH MONITOR

by

JORGE DAVID HERRERA GONZALEZ

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2007

ACKNOWLEDGEMENTS

First I want to thank my father who's always encouraging me to find new goals and to fight for them until I have had accomplished them. To my mom who's since the very beginning of this studies was the most happy with that idea. To my wife who support me and help me every time I have to employ family time in my studies. To my son Nicolas and daughter Andrea, who help me to relax when all the work from the university and my company press me, and for giving me the most important reason to improve every day.

To David Levine for help me since I arrived to UTA and for bringing always a happy and positive ambient in the university.

ABSTRACT

DISTRIBUTED PREDICTIVE

HEALTH MONITORING

Publication No. _____

Jorge David Herrera Gonzalez, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: David Levine

According to the National Center for Health Statistics, the major causes of death in the US are (per year) heart diseases with 654 thousand cases, followed by cancer with 550 thousand cases and stroke with 150 thousand cases. In this project we propose, design, implement and test a prototype of a system that promises to improve the level of detection and prediction of several diseases and improves the quality of life of patients by utilizing non-invasive wireless medical sensors, a small data collector and remote servers.

This system is composed of a set of wireless medical sensors, a nearby (close-proximity) PDA (personal digital accessory) or cell phone used for data gathering and short-term analysis, and a large server used as a data repository and longer-term

analysis and visualization center. Medical sensors are used to monitor cardiac function (ECG), as well as other medical conditions, while the "local" device (for example a cell phone) collects sensor information, does a preliminary analysis (looking for simple anomalies) and collects and sends information to a larger server. The server looks for larger (days or months) trends, detecting unusual conditions as well as creating visual displays of the patient's data.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Definition of the problem

Medicine has received special attention in the US in the last years, partially due the changing of the age distribution of the population. In the US, the percentage of people older than 65 was 4% in 1900, 13% in 1990 and is expected to be 17% in 2010.[1] Because aging increases the probability of the appearance of medical problems and the necessity of care, the number of patients is increasing while at the same time the number of people attending to them is decreasing. Elderly people also often need some form of on-going care, sometimes only to check that he/she has taken medicine on time or a continuous check of one or more vital signs. On the other hand, they do not want to loose their independence, but a lack of care would put them at risk.

For all people, being sick is a complex problem. Completing all the steps necessary for a diagnosis are normally high cost, painful and imprecise.

Nowadays it is common that nurses start the analysis and diagnosis process by taking the patient vital signs and comparing them with the data taken the last time the patient visited. At this moment, the doctor receives only three pieces of information; the description of the symptoms, the data of the current visit and the data of the last visit.

---

[1] Wan He, Manisha Sengupta, Victoria A Velkooff and Kimberly A de Barros (December 2005) 65+ in the United States: 2005. Retrived October 3 2007 from http://www.census.gov/prod/2006pubs/p23-209.pdf

The doctor cannot know what happened between these points in time. If the problem shows up only in certain situations, it is possible that the cause of the problem will not be detected by the devices used by the doctor initially, so the doctor will need to develop new strategies to record more data out of the office or force situations in the laboratory to catch the source of the problem. For example, if the patient has a heart problem that shows symptoms only under stress, the physician will need to create stress to obtain more information. In cases such as sleep apnea the patient should use a device to record data for a long time period and a specialist must interpret the data that is generated every second. This data may be very long, so the price for the analysis would be very expensive and there is a chance that the specialist might miss important data.

After a patient receives a diagnosis, the patient is usually checked several more times to check the development of the disease. The patient could stay in the hospital, paying a high cost, pay for private care at home, or frequently visit a hospital or a laboratory.

There are illnesses, such as asthma or pneumonia that require that the patient is checked periodically. If the patient is not very ill the doctor can send the patient home and ask that he/she come back at some later point in time; but if the patient gets worse during this time, the doctor will not know and the patient could be at a higher risk.

Today doctors and nurses use several devices to measure vital signs, height, weight and other information of the patient, and his/her symptoms, to generate an initial

diagnosis. Based on this, a doctor uses more specialized devices to get the necessary information to make a more detailed diagnosis.

Some of these devices have been improved with the goal of being less invasive and to take advantage of the advances in microelectronics and particularly the introduction of low cost wireless connections such as Bluetooth technology. These new devices can record large amounts of data covering different situations the patient is involved in during his normal live. But for the doctor it is difficult to analyze all of this information due the large amount of data. Many devices can alert a user o a doctor when the data is out of bounds but can only perform simple analysis of the data. For example, if the heart rate is high the device can not differentiate whether the patient is exercising or resting.

This thesis proposes a system that integrates medical devices, non-medical devices such as GPS, accelerometers and sensors, a PDA and a server with the goal of giving better information to the doctor, displaying only when the data is relevant, and allowing the doctor to check medical data online and in real time to assess the patient's medical status. Because the system takes advantage of wireless technologies, the patient is unplugged, so he can perform normal life activities and stay remote from the doctor's office.

In the system proposed here, medical devices such as electrocardiogram (ECG), electroencephalogram (EEG), blood pressure sensors, Temperature sensors, oxymeters, and non-medical devices like accelerometers, Global Positioning System (GPS) and other sensors are connected wirelessly to a PDA that collects analyses the data. If the

PDA detects some data that is out of the pre-set limits, it can alert the patient and the doctor. It is important here that the PDA can analyze the data of one device in conjunction with the data of other sensors; for example, the PDA can detect that the patient is doing exercise using accelerometer data and change the detection limits of other devices such as heart rate or ECG sensors.

The data obtained by the PDA is sent to the server, where it is saved in a database. The server can perform a more complex analysis such as comparing data from different dates or run a special analysis of the data that the doctor suggests.

The doctor's interface to the system could show only the data that the server has detected and flagged as important to analyze, so the doctor will save time and reduce and the probability that he misses some important data changes.

This system should facilitate and improve the diagnosis task of the doctor while the patient experiences few restrictions to his freedom.

CHAPTER 2

MODERN MEDICAL DEVICES

New medical devices have been developed with the goal of being less invasive and to take advantage of the advances in microelectronics and especially the introduction of low cost wireless connections like bluetooth technology. Some notable devices are:

2.1 ECG Recorder

This device records a few seconds of ECG data and sends them through Bluetooth technology to a computer or a PDA[2]. The ECG records the levels of electric changes close to the area of the heart during a period of time. These changes are related to the depolarization and repolarization of the heart. With this data we can obtain the heart rate, amplitude of the QRS complex and some other data useful to detect heart diseases. In this implementation we use an early beta version of this product.

---

[2] CORSCIENCE (2007) CorBELT System. Retrieved December 6, 2007 from
http://www.corscience.de/index.150.en.html

Figure 2.1 ECG Recorder

2.2 Blood pressure sensor

These devices measure arterial pressure[3]. They can be useful to diagnose hypertension (high level of arterial pressure). Persistent hypertension can cause heart attacks and renal failure. Sometimes people are diagnosed erroneously with hypertension because of the level of anxiety at the moment the doctor tests the patient. Measuring this data at home would decrease this false positive error.

---

[3] CORSCIENCE (2007) Bluetooth Blood Pressure Meter. Retrieved December 6, 2007 from http://www.corscience.de/index.158.en.html

Figure 2.2 Blood pressure sensor

2.3 Pulse Oximeter

This device provides the oxygen level of the blood as well as the pulse rate[4]. This data can help people with pulmonary problems such as asthma or pneumonia. The system can also detect a low level of oxygen related to a heart problem.

Figure 2.3 Pulse Oximeter

---

[4] CORSCIENCE (2007) Pulse Oximeter OxiPen. Retrieved December 6, 2007 from http://www.corscience.de/index.72.en.html

2.4 Body Temperature Sensor

The body temperature tells the doctor of the presence of pyrogens produced by the immune system[5]. This data is usually related to infections, diseases or adverse reaction to drugs.

Figure 2.4 Body Temperature Sensor

2.5 Glucose level Sensor

This type of device is under research. It will bring valuable information for the analysis of and response to low or elevated glucose levels[6].

---

[5] Wikipedia (2007) Thermoregulation. Retrieved December 6, 2007 from
http://en.wikipedia.org/wiki/Thermoregulation
[6] David C. Klonoff (May 10, 2007) Chapter 21. MONITORING TECHNOLOGIES- CONTINUOUS GLUCOSE MONITORING, BIOMARKERS OF GLYCEMIC CONTROL, ARTIFICIAL PANCREAS Retrieved December 6, 2007 from http://www.endotext.org/diabetes/diabetes12/diabetes21.html
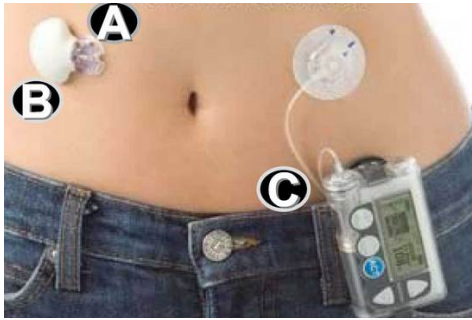
Figure 2.5 Glucose level Sensor

2.6 Spirometer

The Spirometer measures the amount and speed of the air inhaled and exhaled[7].

This data can help a doctor detect possible asthma or pulmonary fibrosis.

Figure 2.6 Spirometer

---

[7] CORSCIENCE (2007) Bluetooth Peak Flow Meter. Retrieved December 6, 2007 from http://www.corscience.de/index.154.en.html

2.7 Advantages and disadvantages

New devices produce digital data that can be used by computers. This allows programmers to create new utilities to help doctors and patients. Using this data programmers can produce new algorithms and interfaces to assist doctors. Bluetooth and other wireless technology replaces the cables used to connect the devices and the doctor's interface. This helps the patient feel more comfortable and allows him/her to perform normal tasks that he/she could not do if required to be connected using cables.

But the experience with these devices has shown us that there are some points that may be improved. While having a larger amount of data is very good in that it contains more information that can be used to perform diagnosis, it can also sometimes have drawbacks for the doctor. It could be a very hard task to analyze all this data, and because it is so large, the doctor could miss some important detail.

The devices used today work independently so they can only alert when data is out of bounds or perform other low complexity analysis. For example, the ECG alone can not differentiate if the patient is exercising or not.

CHAPTER 3

PROPOSED SOLUTION

3.1 Overview

Figure 3.1 General System

The goal of the proposed system is to build a system that can use multiple

sensors and link them wirelessly to obtain local analysis capabilities in real time as well

as a link to the doctor's office and advanced analysis on the doctor's computer using all the data accumulated over time.

All medical devices are connected to a PDA using Bluetooth or WI-FI. They send information continuously or each minute to the PDA. The PDA analyzes the data continuously looking for data out of normal parameters. If the PDA finds something abnormal, it can send a message to the doctor or alert the patient. The PDA is connected to a server and sends all the sensor values received from the medical devices. If the PDA can not connect to the server, it stores the data temporarily until the connection can be initiated; then the PDA first sends all the old data and after the file is empty, the PDA sends the fresh data to the server. All the data sent to the server has a time value that identifies when the data was generated by the medical devices.

The server stores the data received in a database and analyzes it. If it finds something interesting it sends the information to the doctor.

When the server has received enough data, then the specialist can start analyzing it. The interface to the system allows the doctor to check data that could show an anomaly. This interfaces buttons; one for each analysis the doctor wants to check.

When the specialist finds a particular minute's data that he wants to check, he can select this minute and the interface will show him all the values recorded during this minute and will allow the doctor to move to check other minutes near this one.

A general system should integrate:

- Medical devices: Devices that capture medical information and send it to a personal collector.

- Positional devices: Devices that detect the position and the movements of the person.

- Personal collector: PDA, cell phone or personal device that collects the medical and non-medical information, records it and sends it to a Server. If it is possible, the PDA performs some analysis.

- Server: Receives data from the personal collectors, stores and analyses it looking for medical problems based on immediate data and comparison of data over time. Sends alerts to the medics via email, SMS, or other channels.

- Web server: Allows the doctors to connect from their offices and read all the patient data.

3.2. Non-Medical devices

The medical devices are the main source of data a doctor wants to analyze, but sometimes this data are external conditions such as the ambient temperature, the altitude, or what the patient is doing or has done; for example: is he/she exercising? Has he taken his medicine? When? Has he eaten? All of these questions can be answered using information from an accelerometer and some ingenious but simple sensors installed in the medicine dispenser or in the kitchen.

In case of an emergency the system could supply the location of the patient from GPS information.

3.3. Low Range Collector

To collect sensor data we need a small device that can be programmed. It could be a PDA[8] or a cell phone. This device should have at least Bluetooth and WI-FI capability. If we want the device to send the data through the Internet, the device should have GPRS[9] and a service that allows the cell phone connects to computers using TCP/IP. Nowadays, we can use a cell phone with Java programming capability or a PDA that supports C++ or similar languages. This device should connect using wireless technology to the medical and non-medical devices and ask them for medical information. Then, the device should send this information to the server using TCP/IP or GPRS.

Figure 3.3 Low Range Collector

---

[8] Wikipedia (2007) Personal digital assistant Retrieved December 6 2007 from
http://en.wikipedia.org/wiki/Personal_digital_assistant
[9] GSM Association (2007) GPRS Plataform Retrieved December 6 2007 from
http://www.gsmworld.com/technology/gprs/index.shtml

Depending on the complexity of the operations necessary to detect some anomalies, the PDA may alert the patient or the doctor immediately.

3.4. Medical Server

A server collects and analyzes the data. The data collected and the result of the analysis is stored in a Database. After that, the server provides the doctor with data using web browsers and send alerts to cell phones using SMS, email, or other communications technologies.

Figure 3.4 Long Range Collector

CHAPTER 4

PROTOTYPE SYSTEM

4.1 Description of the Prototype System

To prove the viability of a project with characteristics similar to the one previously described we created a system to collect the ECG data of a person using an ECG medical sensor device, a PDA cell phone to collect the ECG information, a wireless router to connect the PDA and the server and the software necessary for the PDA and the server.

The ECG recorder sends segments of one second to the PDA using Bluetooth technology. This data is compressed and has a checksum to ensure the data is identical in the PDA.

At the beginning, the PDA should be paired with the medical device. After that the PDA can use a serial port to connect with the medical device and works as the master in this connection. Then, the PDA asks for information every second. This ECG data is analyzed in the PDA and if the data crosses the normal limits the device can alert the patient. The data is then sent to a server or saved in a file until the PDA connects to the server.

The server analyzes the data and stores all the ECG data and the results of the analysis. For this experiment we did not implement the software to send an alert to the medic but rather produced it directly on the device; this code would not be dificult to

develop. We also did not use a web server. Rather, the data and the doctor's analisis interface were done using a Win32 program on the server.

The language used for the PDA was C++ and the compiler was Microsoft Visual Studio 2005. For the server interface we used Delphi 7 and the database was developed using MS-SQL.

CHAPTER 5

ELECTROCARDIOGRAM

5.1 Definition

An electrocardiogram or ECG is the data and graphic representing the levels of electric changes close to the area of the heart during a period of time[10]. These changes are related the depolarization and repolarization of the heart.

Because this research is not medical, we will not include too many details of the medical aspects of the ECG here, but will include an appendix with links to web pages with more medical details.
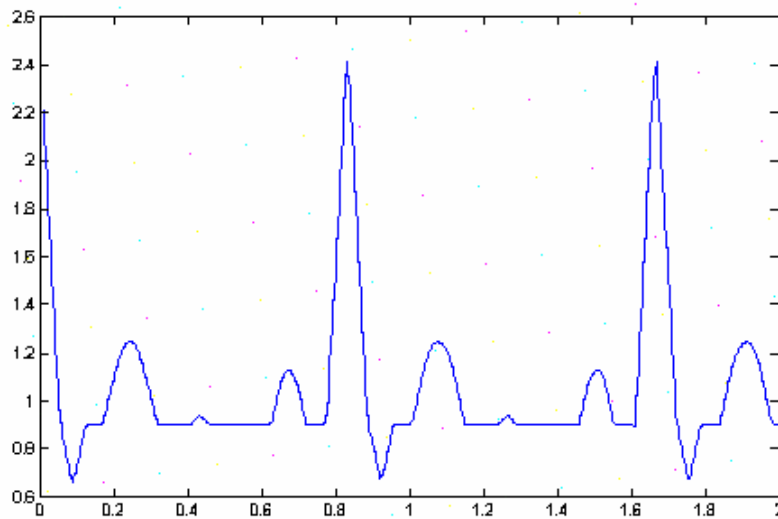
Figure 5.1 Simple ECG

---

[10] American Heart Association (2007) **Electrocardiogram (EKG or ECG)** Retrieved December 6, 2007 from http://www.americanheart.org/presenter.jhtml?identifier=3005172

Figure 5.2 shows the values of a normal ECG curve, and the normal distance

between the important points[11].
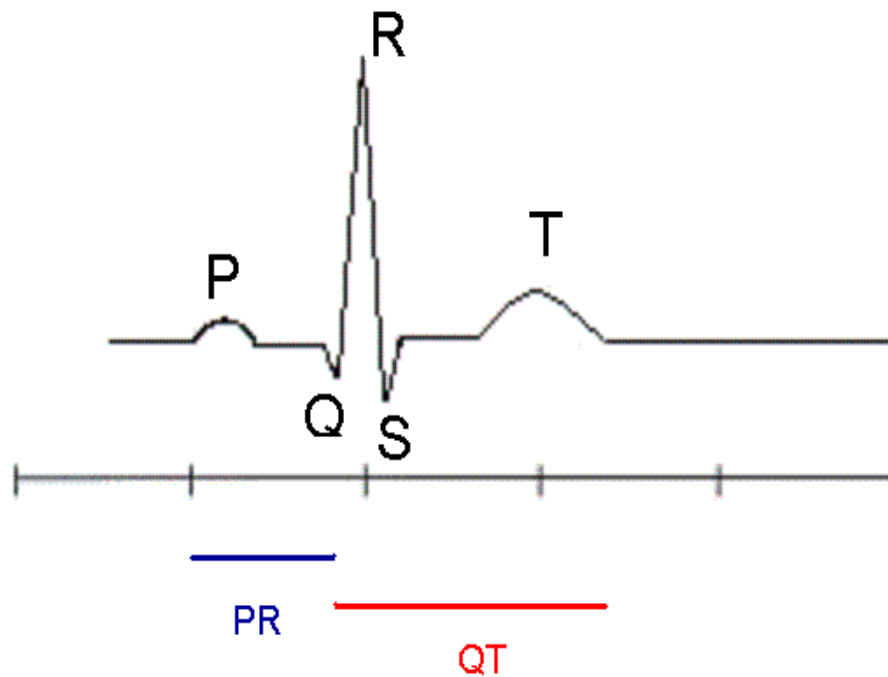
P wave: 0.08 - 0.10s
PR: 0.12 - 0.20s
QRS: 0.06 - 0.10s
QTC = QT / SQR(RR)
QTC < 0.45

Figure 5.2 Schematic representation of normal ECG

---

[11] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from
http://en.wikipedia.org/wiki/Electrocardiogram

While Dr. Willem Einthoven was not the inventor of the ECG, he was the person who assigned the names to the important points of the graphic and defined the normal shapes and distances over time. From this base, he described the graphics resulting from several diseases. All this work helped Dr Willem Einthoven to earn the Nobel prize in 1924[12].

5.2.1 Heart Rate

The activity represented in Figure 5.2 is repeated by the heart continuously. The normal resting rate is between 60 and 90 cycles per minute. Less than that is called bradycardia and higher rate is tachycardia[13]. In general, the repetition of the cycles should be similar but not necessarily identical.

5.2.2 P wave

The P wave is the result of the depolarization of the SA node (Sinoatrial node, pacemaker site) through the atria[14].

The normal height of the P Wave is 2.5mV and its duration is 0.11 seconds. More than 0.3 mV is a sign of a right atrial enlargement and a wider wave could be a sign of a left atrial enlargement[15]. (refer to the Graphic 5.2)

---

[12] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #History
[13] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram
[14] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #P_wave

5.2.3 PR Interval

This segment starts at the beginning of the P wave and finish at the beginning of the QRS complex.

The normal length of this segment is between 0.120 and 0.2 seconds. For this research this data is very important because it implies the possibility of several cardiac problems. The following are the most important and detectable for this version of the system:

- Time longer than normal may indicate a first degree block caused by slowed conduction.

- Shorter may indicate pre-excitation.

- Variable interval may indicate a type II second degree block or the Wenckebach phenomenon.

5.2.4 QRS complex

This part of the ECG is the result of the depolarization of the ventricles. The height of this curve is usually higher due to the larger size of the muscle mass of the atria. This graphic is different in the sense that it is spiked, but it can still contain different shapes. In Figure 5.4, we can see different configurations of this segment.

---

[15] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from
http://en.wikipedia.org/wiki/Electrocardiogram
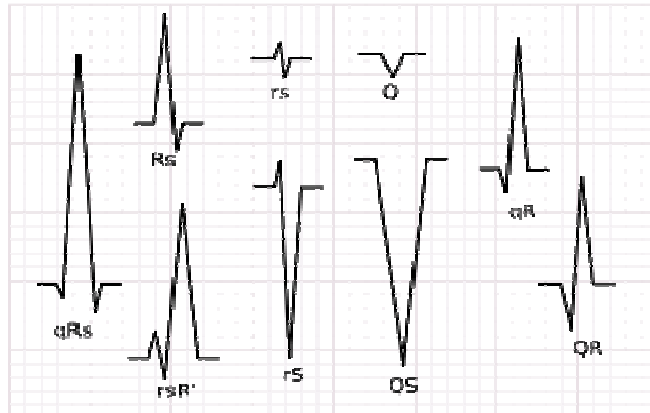
Figure 5.3 Possible shapes of the QRS segment

The normal length of this segment is between 0.06 and 0.10 seconds[16]. There are several aspects of interest in this segment but most of them need more than two nodes. For this thesis the most important was the change of the amplitude over time because it may signal a pre-infarction. Also, the relation between the height of Q and R is important because if Q is greater than 1/3 of R this might indicate a myocardial infarction. A myocardial infarction could also be signaled by a detection of a Q longer than 0.040 seconds.

5.2.5 ST segment

The ST segment starts at the end of the QRS complex and finishes when the T wave starts[17]. This line has a slight upward concavity and its length is between 0.08 and 0.12 seconds. Some anomalies could be detected after analysis of slight differences from the normal behavior but this data is difficult to measure due to the difficulty

---

[16] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

detecting the beginning of the T wave and the fact that noise in the system could be greater than the normal values accepted for ST. The only case the system built here will detect is when the ST segment is taller than the T wave and at least ¼ R. We will advise of a possible myocardial infarction, noting that the probability of false negatives and positives is between 15-30%.

5.2.6 T wave

The T wave shows the ventricular repolarization. This curve is normally curved to same direction as the largest deflection of the QRS complex. A spiked curve may indicate a Hyperkalaemia problem[18].

5.2.7 QT interval

The time elapsed between the end of the QRS complex and the end of the T wave is important to detect two possible problems. The normal time is about 0.4 seconds but this time changes according to the heart rate, so it is necessary to calculate the relation between QT time and the time of a cycle.

The corrected QT is equal to QT/SQR(RR), where RR is the time for a cycle. The normal value of QTc is 0.39 +/- 0.004 seconds (this value could be a little larger in women) A value larger than 0.5 seconds may indicate poisoning with tricyclic antidepressants, congenital QTsyndromes, hypocalcaemia, and toxicity from other

[17] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from
http://en.wikipedia.org/wiki/Electrocardiogram #ST_segment
[18] Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from
http://en.wikipedia.org/wiki/Electrocardiogram #T_wave

drugs. Other causes could be hypothermia, head injury, acute myocardial infarction and hypertrophic cardiomyopathy.[19]

5.2.8 U wave

The U wave is a very small wave after the T wave. If the system detects a big U wave it will advise about a possible first or second degree AV block.

---

[19] Lara Hopley and Jo van Schalkwyk. (2006/07/24) ECG scribbles. Retrived October 5 2007

CHAPTER 6

SYSTEM COMPONENTS

The system developed in this thesis is composed of three electronic devices. A Medical Device (Corbelt ECG), a PDA with Bluetooth and WLan using a Microsoft operating system, and a personal computer working as a server.

6.1 Medical Device

The medical device used was a Corbelt ECG recorder[20]. This device is a personal ECG recorder that weights only 150g and is very comfortable to wear. It has two Dry electrodes and transmits ECG data at 200Hz using 12 bits of resolution.

Figure 6.1 Corbelt ECG Recorder

This device needs only one AA battery and uses Bluetooth technology to send the data, so the patient is absolutely free of any cables.

The device is discoverable all the time and should be paired with the PDA using a fixed password. After the Corbelt and the PDA are paired, the PDA can start the

connection and send requests for data to the Medical Device. (The version used was the first beta)

6.2 PDA

The main function of the PDA is to record the medical data and send it to the server when the PDA is within range. Therefore the PDA should have Bluetooth capability, storage memory space and a way to connect to the server. In this work we choose a WLAN connection, but it could be connected in other ways that guarantee that the user can eventually transfer the data from the PDA to the server. One possibility for this if we want the user to be monitored all time and if the patient needs to be outside the LAN would be to connect though GPRS.

Figure 6.2 HP iPAQ 6325

For this work we choose a HP-IPAQ 6325 using Pocket PC 2003. To store the ECG data the PDA uses a ScanDisk memory of 64 MB. The executable generated for the PDA was produced with Visual Studio 2005 using C++.

---

[20] CORSCIENCE (2007) CorBELT System. Retrieved December 6, 2007 from http://www.corscience.de/index.150.en.html

6.3 Server

The server receives data using TCP IP through WLAN. For this thesis I choose a DELL Inspiron 6400 Laptop with 1 GB of ram and a wireless card. This computer uses Microsoft windows XP.

Figure 6.3 Lap working as server

The code for this server was created using C++ and was compiled with Visual Studio 2005.

6.4 WLAN

The PDA and the server are connected using a wireless LAN. To simulate one I choose a Linsys WLAN Router WRT54G.

Figure 6.4 WLAN Router WRT54G

This configuration can create a common home network or using several of these we can simulate a system installed in a hospital. It is possible to connect the server

through this router to the Internet and work as an Internet server so the PDA can connect to the server more often and from a wider range using a GPRS connection.

CHAPTER 7

PROTOCOLS

Studying the project from bottom to top, we will now present the protocol to connect the medical device to the PDA through Bluetooth and the PDA and the Server using TCP/IP.

7.1 Corbelt - PDA Connection

The Corbelt ECG Recorder connects only using Bluetooth. The protocol and the device are produced by CORSCIENCE. (www.corscience.de). The version of the protocol this device uses is Rev. 3.2 12/2006.

To connect the Corbelt and the PDA, it is first necessary to pair the device and the PDA. Then the PDA can open the serial port assigned for this device.

The PDA uses two functions, send_Corbelt and receive_Corbelt to ask for ECG data. These two functions encode or decode the data in packets that the Corbelt understands.

7.1.1 *Corbelt Packet structure*

The packet structure is used for both outgoing and incoming packets. These packets have two security controls. Each packet has a consecutive identification number and a checksum that ensures integrity.

Start flag	packet-number	command	payload-length	payload	CRC	end flag
0xFC	1 byte	2 bytes	1 byte	x bytes	2 bytes	0xFD

Figure 7.1 Packet Structure

7.1.2 *Packet stuffing*

OxFC and 0xFD can not appear in a packet except at the beginning and at the end so the PDA and the Corbelt should check if one of them occurs in the data and translate it. OxFC should be changed into two bytes, 0xFE 0xDC. OxFE into 0xFD 0xDD and 0xFE into 0xFE 0xDE. The checksum should be calculated before this change.

7.1.3 *Commands Used*

The following is the list of commands used in this project. There are more in the description provided by CORSCIENCE.

Command	Number	Direction Corbelt-PDA	Description
reset	0x00FF	←	resets the Corbelt
Acknowledge	0x0200	↔	Single packet with distinct packet number Is acknowledged as correct
self test	0x0600	→	. results of the self test
transmit data	0x070X	→	The defined data is transmitted as payload. The receiver knows the packet structure due to the device ID and the preceding configuration command.
transmit recorded data	0x071X	→	In the device buffered data is transmitted.
Request	0x0800	←	With this command another command can be requested.

Figure 7.2 Commands Used

7.1.4 Details of Commands Used

7.1.4.1 Acknowledge:

    Command: 0x200

    Length: 1

Payload: Byte 0: packet number of the packet, which should be acknowledged

7.1.4.2 Self test:

Command: 0x600

Length: 2

Payload: Byte 0 and 1: short int with an error value. 0 means no error.

*7.1.4.3 T*ransmit data:

Command: 0x700. Indicates the beginning of ECG data transmission.

Length: 10

Payload: Byte 0 and 1: Event marker.

Byte 2 – 10 : Time of the events

Command: 0x701. Transmission of a packet with 107 points of the 200 per

second.

Length: 216

Payload: Byte 0: format ID: 0 means error.

Byte 1 – 214: Data of 107 points of 12 bits.

Byte 215: heart frequency. It never worked.

Command: 0x705. End of the Transmission.

Length: 0

Payload:

*7.1.4.4 T*ransmit recorded data:

Command: 0x710. Indicates the beginning of recorded ECG data

Length: 10

Payload: Byte 0 and 1: Event marker.

          Byte 2 – 10 : Time of the events

Command: 0x711. Transmission of a packet with 107 points of the 200 per

          second (Previously recorded).

Length: 216

Payload: Byte 0: format ID: 0 means error.

          Byte 1 – 214: Data of 107 points of 12 bits.

          Byte 215: heart frequency. It never worked.

Command: 0x715. End of the Transmission of the recorded data.

Length: 0

Payload:

7.1.5 C++ functions that use Corbelt commands

The connection between the PDA and the Corbelt is realized using the BTConnexx object.

```
class BTConexx
{
public:
    int    Connect(long baud_rate,LPCWSTR port_name);
    int    Send_Packet(int command,unsigned char *data, int size);
    int    Receive_Packet(uchar *data, int *command,int Max_size,int *npaq);
    int    Disconnect(void);
protected:
    HANDLE  serial_port;      /* Handle to the serial port */
    int    Packet_num;
};
```

7.1.5.1 Connect:

This method opens a serial connection to the port assigned after the pairing operation using the function:

serial_port = CreateFile(port_name, GENERIC_READ | GENERIC_WRITE, 0, 0, OPEN_EXISTING, 0, 0);

7.1.5.2 Disconnect:

This method closes the serial connection using the function:

CloseHandle(serial_port);

7.1.5.3 Send_Packet:

This method sets up a packet and sends it. It creates a new packet using the command and the data passed in as parameters, the number of the packet and a cheksum calculated for this packet. Then the packet is sent using the serial connection.

(The lines of code that validate the data and results after calling other function has been removed from the code segment presented here in order to make the code more readable).

```
int    BTConexx::Send_Packet(int command,unsigned char *data, int size)
{
char *s,*t;   //memory for the packet. S is before change the characters grater
              than 0xfb. T is the packet after the change.(packet stuffing)
 int copied=0,x;
 unsigned long bytes_written;
 unsigned short chk;



 s=(char *)malloc(size*2+20);
 t=(char *)malloc(size*2+20);
```

```
s[0]=Packet_num;  //packet number

s[1]=command%256;    //setup the command

s[2]=command/256;

s[3]=size;

if (size)

    memcpy(s+4,data,size);   //copy the payload

;


t[0]=(char)0xfc;        //header

copied++;             //numb bytes copied



chk=Calc_Chk(s,size+4);

s[size+4]=chk%256;     //checksum at the end

s[size+5]=chk/256;



for (x=0; x<size+6; x++) { //change the characters grater than 0xfc

    if ((s[x]==0xfc)||(s[x]==0xfd)||(s[x]==0xfe)) {

        t[copied]=(char)0xfe;

            copied++;
```

```
            t[copied]=(char)(s[x]^0x20);//transform bytes similar to headers

            copied++;

        }

    else {

            t[copied]=s[x];   // less than 0xfc

            copied++;

        }

}


t[copied]=(char)0xfd;   //finish signal

copied++;




if (WriteFile(serial_port, (void *)t, copied, &bytes_written, NULL)!=0) {

        Packet_num++;  //write the new packet

        if (Packet_num==252)  //the maximum packet is 251

                Packet_num=0;

        ;

}

else {


        //error
```

```
        }

        free(s);

        free(t);

        return(0);



    }
```

7.1.5.4 Receive_Packet:

This method receives data starting where it finds a byte with the value 0xFC and until 0xFD. The function checks the integrity of the packet using the checksum and compares the number of the packet received against the last packet number received plus one. After all the checks, the function looks for the byte 0xFE and converts the bytes back to the the initial value before the octet stuffing. After this the function can extract the command number and the payload.

The function GetData converts the packet from octet stuffed to normal data.

(The lines of code that validate the data and results after calling other function have again been removed from the code fragment included here to make the code more readable).

```
int GetData(unsigned char *s,unsigned char *data,int count)

{

        int copied=0;

        int x;
```

```
        for (x=0; x<count;x++) {


            if (s[x]==0xfe) {

                x++;

                data[copied]=s[x]^0x20;

                copied++;

            }

            else {

                data[copied]=s[x];

                copied++;

            }

        }

        return(copied);

}


//*****************************************


int    BTConexx::Receive_Packet(uchar *data, int *command,int Max_size,int *npaq)

{

 int count=0,size,n;

 unsigned long bytes_received;

 unsigned char *s;
```

```
*command=0;

s=(unsigned char *)malloc((Max_size+20)*3);



int done=0,times=0;

//look for the initial header

while (!done) {

        if (ReadFile(serial_port, (void *)(s+count), 1, &bytes_received, NULL)==0) {

                if (s[0]==0xfc)

                        done=1;

                ;

        }

}

s[0]=0xfc;

int rep=0;

data[0]=0;

while (count < Max_size*2) {

        if (ReadFile(serial_port, (void *)(s+count), 1, &bytes_received, NULL)==0) {

                //error(L"error reading");

        }

        if (s[count]==0xfd)//read until end mark
```

```
        break;

    ;

        count++;

}


size=GetData((unsigned char *)(s),(unsigned char *)data,count);
if (size<6) {

    free(s);

    return(0);

}
{

 int nsize;

 unsigned int z;

 unsigned int coming_chs;

 unsigned char *p;


 z=Calc_Chk((char *)(data),size-2);

 nsize=(size-2);


 p=((unsigned char *)data)+nsize;

 coming_chs=(unsigned int)(p[0])+(((unsigned int)p[1])*256);

 if (z != coming_chs) { //check integrity
```

```
    free(s);

    return(0);

 }

}

 *command=data[1]+data[2]*256;//get the command

 *npaq=(unsigned int)((BYTE)data[0]);

 if (size>6)

    memmove(data,data+4,size-6);  //delete the heaader

 ;

 free(s);

 return(size-6);

}
```

7.1.6 High Level function to collect the data from the ECG recorder

To get the data from the ECG recorder, the PDA opens the port, sends a packet to check the status of the medical device and checks in an infinite loop for data packets. When the PDA receives commands 0x701 or 0x711, the PDA stores this data in a circular buffer so the system can analyze the data or send it to the server later. If the PDA does not receive packets for a while, it sends a packet with a petition for data. If the connection fails the PDA connects again. The following is a summary of the thread collecting the data:

```c
*((short int *)(p))=(short int)0x0600; // Check ECG status

BT->Send_Packet(0x800,p,2);


do {

    p[251]=0;

    size=BT->Receive_Packet(p, &command,300,&paq_num);

    if (!command) {

        // error(L"no command");

    }

    else{

        WCHAR ss[300];

        Log("arrived  the  command=%x  size=%d  paq=%d",command,size,paq_num);

        switch(command) { //wich command is arriving

            case 0x700: //ECG started sending data

                getCORtime(p+2, &year,&month, &dayofmonth, &hour,&min,&secs);

                Log("time=%d-%d-%d %d:%d",month,dayofmonth,year,hour,min);

                Log("event=%x",*((short int *)p));

                SaveData(p,0x700,20);

                p[0]=0x01;  p[1]=0x07;  // Acknowledge please continue

                BT->Send_Packet(0x800,p,2);

                break;

            case 0x710: //ECG started sending recorded data
```

```
                    getCORtime(p+2, &year,&month, &dayofmonth, &hour,&min,&secs);

                    Log("time=%d-%d-%d %d:%d",month,dayofmonth,year,hour,min);

                    Log("event=%x",*((short int *)p));

                    SaveData(p,0x710,20);

                    p[0]=0x11;  p[1]=0x7;  //*((short int *)(p))=(short int)0x0711;


                    BT->Send_Packet(0x800,p,2);

             break;

            case 0x701:  //ECG data comming

                    SaveData(p,0x701,216);  //store the data

                    p[0]=paq_num;

                    BT->Send_Packet(0x200,p,1); // Acknowledge please continue

             break;

            case 0x711:

                    SaveData(p,0x711,216);

                    p[0]=paq_num;

                    BT->Send_Packet(0x200,p,1); // Acknowledge please continue

             break;

            case 0x705:  //end

                    p[0]=paq_num;

                    BT->Send_Packet(0x200,p,1); ); // Acknowledge

                    {
```

```
            unsigned char p[300]={0x00,0x07}; //0x700

              BT->Send_Packet(0x800,p,2);  //Please keep sending ECG data

            }

        break;

      case 0x715: //end

          p[0]=paq_num;

          BT->Send_Packet(0x200,p,1); ); // Acknowledge

        break;

      }

    }

 } while(contin);
```

7.2 PDA - Server Connection

Connecting the PDA and the server is much simpler than connecting the PDA and the medical devices. The code uses the normal TCP/IP protocol and normal file C++ APIs to temporarily store the data or connect to the server and send the stored information.

7.2.1 Starting the connection

The PDA tries to connect to a fixed IP address and port. If the connection fails, the packets received from the medical device are stored in a file until the connection succeeds.

The file that keeps the packets has a header containing two integers. The first integer is the number of packets extracted from the file and the other integer keeps the number of packets inserted into the file. When the system starts, both numbers are 0 and the file is small. If the PDA can not connect to the server the file grows larger. When the PDA connects to the server, the PDA reads all the packets stored (FIFO) and send them to the server. Then the file is deleted and created again with both integers set to zero. At this moment the PDA starts sending the packets as they are received from the medical devices.

The PDA initially sends some bytes containing the name of the owner of the PDA (the patient), version number and some bytes that identify the source of the connection.

7.2.2 *Sending packets*

After the connection is established, the PDA sends the number or code of the device (medical or not) that is sending the packets, the date the data was received, and the data received by the PDA.

In case another device is integrated, the function that receives the data from this device should generate packets with a different identifier and can then use different size packet. The server code should be upgraded to support this new case and then receives this new kind of packets. This change is very easy and allows the integration of new devices.

CHAPTER 8

ECG DATA ANALYSIS

The data for the devices not always fulfill the requirements of the project. The medical device used for this project has been recently updated. This device only provides the values necessary to reproduce the ECG graphic. Newer versions provide the heart rate and alerts when this value exceeds some limits.

To make the prototype developed here more complete, we created some algorithms to detect not only the heart rate, but also to extract the amplitude of the QRS complex and some data values that are useful for a physician such as QT and PR.

8.1 Real time analysis

After checking the curve (ECG), we realize that the first and easiest point to detect is one in the QRS complex. For this experiment we tried to first detect the S point. The number of points detected over a period of one minute provides the heart rate and every S point provides a clue of where to look for the P and T wave (at least the beginning or the end).

8.1.1 S point detection

The slope is really the main determinant to detecting the QRS complex. The points between R and S keep a high negative slope and the difference of the Y value between these two points is the highest in the graph.

The actual algorithm checks the last 13 points looking for a big initial increment in the Y value or for some points with a high slope value until reaching a maximum, then changing to a very low slope for some points and finally finishing with a positive slope again.
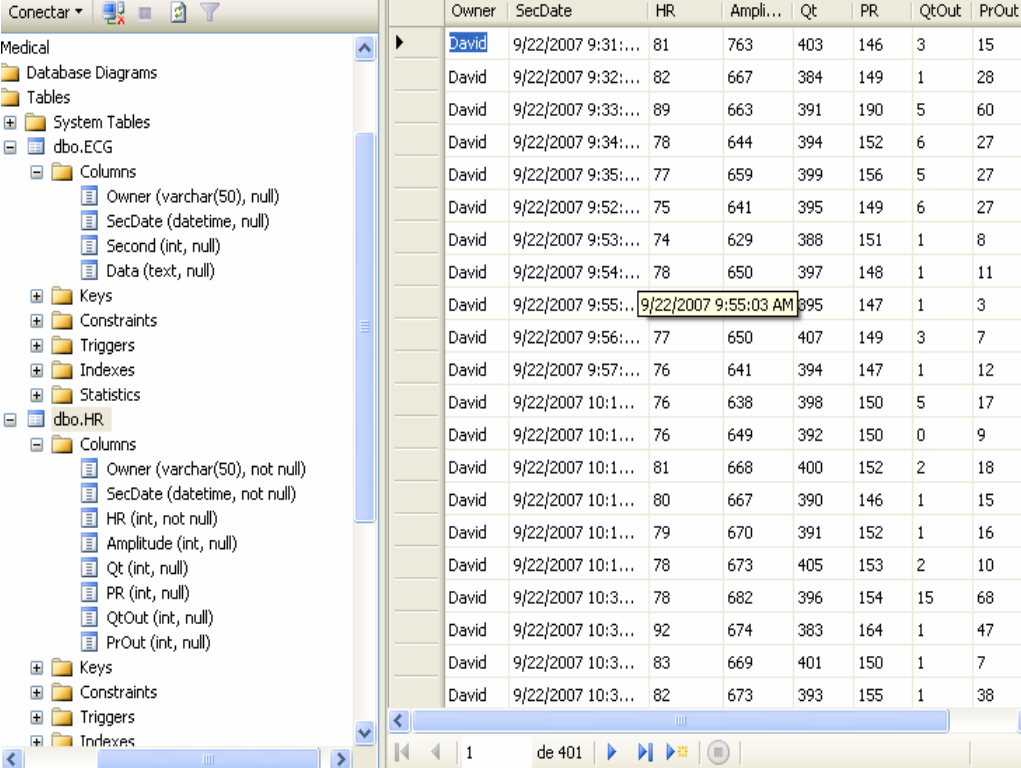
After identifying this S point we look for the highest and lowest point surrounding it to detect the amplitude and look in a similar way as before (using the slopes) in the surrounding relatively flat areas for the locations where the T and P curve should be developed.

Detecting all the points of T and P is sometimes difficult so we focus here on detecting the points necessary to measure the QT interval and the PR interval. These measures can help doctors to detect several diseases.

CHAPTER 9

DATABASE

For a doctor, the online data is very important, but data over time could give other clues of possible anomalies. Having several data segments allows the doctor to compare the behavior of the patient under similar situations and compare it to other individuals' data. If the system is used for several people, the database could bring valuable information to researchers looking for a way to predict diseases or changes in the behavior of the patient's organs.

Figure 9.1 Database

The database for this experiment is composed of two tables. One table, HR, keeps the heart rate, amplitude, QT and PR length, and the times PR and QR had a value exceeding the limits. The other table, ECG, contains the values of every point of the curve for one minute. For this ECG recorder the number of points is 200 per second. Both tables contain columns that keep information about the owner of the data, the date, and the time this data was generated. Every row is generated a minimum of one second apart.

9.1 ECG Table

This table has three keys, The patient ID (Owner), the date in minutes of the minute when the data was collected (SecDate) and the number of second in this minute when the data was collected. The data in this table is 'Data' that contains all the points of the ECG of a second.

9.2 HR Table

This table has two keys, The patient ID (Owner) and the date in minutes of the minute when the data was collected (SecDate). The data in this table is contained in several columns. Every column contains the data collected this minute. The columns are Heart rate (HR), Amplitude (Amplitude), length of QT (QT), length of PR (PR), number of QT segments out of range (QTout) and number of PR segments out of range (PRout).

CHAPTER 10

SOFTWARE INTERFACES

The software developed for this prototype system was not directly driven by costumer demands. On the contrary, this software is a research tool or an indicator of some of the possible data generated by the system.

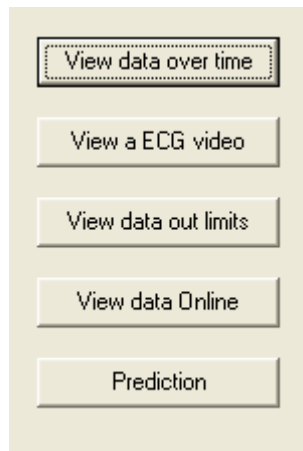The system offers four options in the main menu.

Figure 10.1 Main Menu

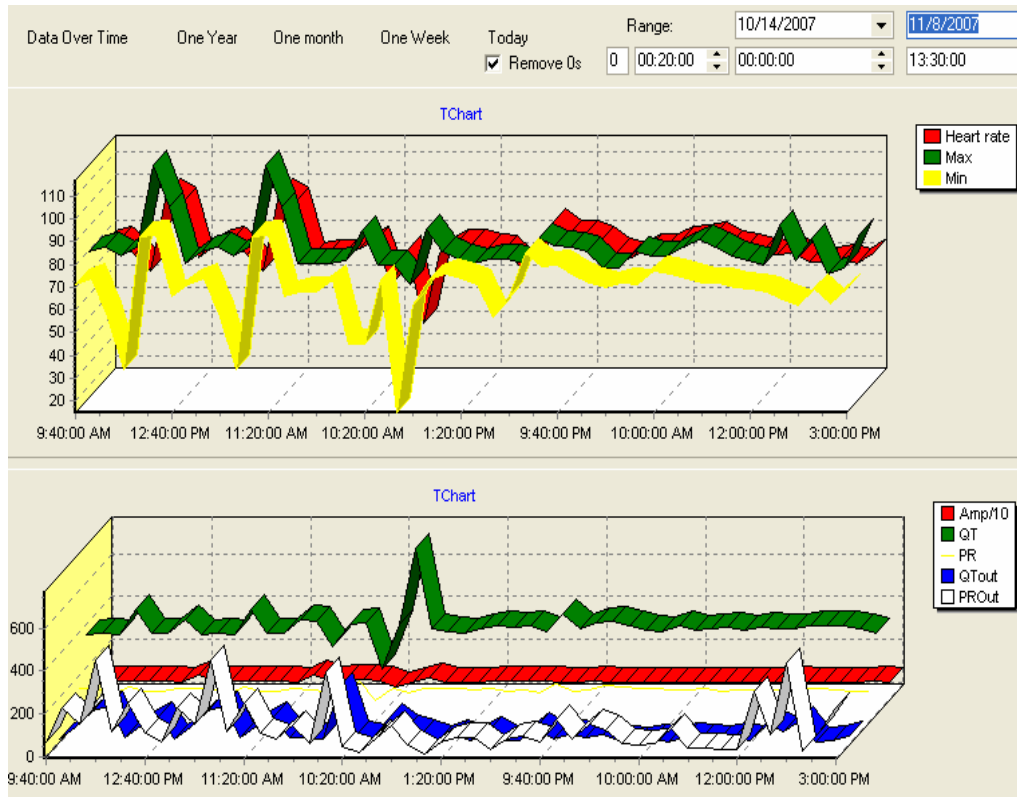## 10.1 View data over time

Figure 10.2 Data over time

This interface allows the doctor compare the values over a period of time. The values can be grouped by periods of fixed length of time. The value of the point would be the average of the values in this period of time.
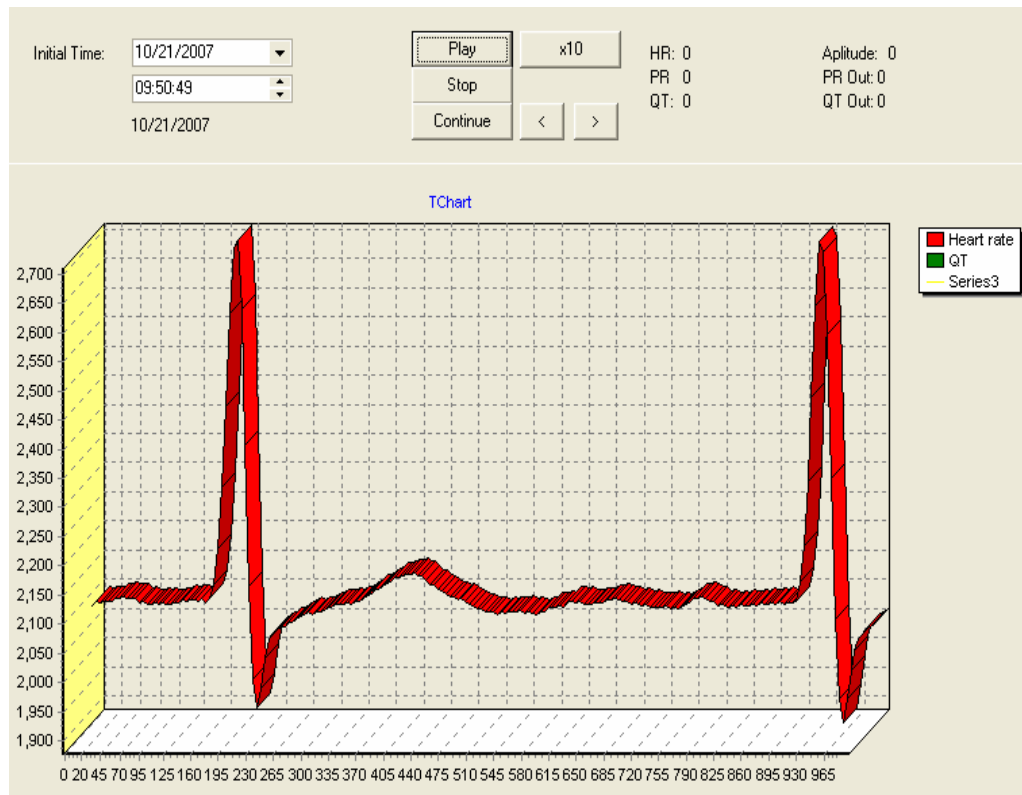
10.2 View ECG Video

Figure 10.3 ECG Video

Using this option the user can see a video of the ECG recorded in the database for a selected minute. The system reads the Heart Rate and the average of the other data for this minute and starts a timer that every second reads the corresponding 200 points and displays them.

10.3 Data Out of Bounds

Figure 10.3 Data Out of Bounds.

Using this interface the user can select a range of time and search for times at which one of the attributes was out of limits. After the user presses a button one interface will show the times at which the data was out of range and allows the user to select an instance to see the video of the ECG generated for this second. In the experiments performed, the majority of the data with values out of range was generated by noise or due to disconnection between the device and skin.

Owner	SecDate	HR	Amplitude	Qt	PR	QtOut	PrOut
David	10/21/2007 10:55:49 AM	34	669	538	174	23	150
David	10/21/2007 11:10:20 AM	117	789	389	135	19	79
David	10/21/2007 11:11:19 AM	94	765	370	136	0	7
David	10/21/2007 11:12:20 AM	92	773	365	135	1	9
David	10/21/2007 11:13:20 AM	94	729	367	135	3	11
David	10/21/2007 11:14:19 AM	94	729	369	134	0	14
David	10/21/2007 11:20:19 AM	93	701	372	132	2	25
David	10/21/2007 11:21:20 AM	93	774	374	133	3	19
David	10/21/2007 11:22:20 AM	93	701	363	128	3	32
David	10/21/2007 11:23:19 AM	94	714	363	139	0	10
David	10/21/2007 11:24:20 AM	94	703	374	132	1	35
David	10/21/2007 11:30:28 AM	94	713	380	140	36	79
David	10/21/2007 11:31:29 AM	99	807	372	138	30	50
David	10/23/2007 10:55:49 AM	34	669	538	174	23	150
David	10/23/2007 11:10:20 AM	117	789	389	135	19	79
David	10/23/2007 11:11:19 AM	94	765	370	136	0	7
David	10/23/2007 11:12:20 AM	92	773	365	135	1	9
David	10/23/2007 11:13:20 AM	94	729	367	135	3	11
David	10/23/2007 11:14:19 AM	94	729	369	134	0	14
David	10/23/2007 11:20:19 AM	93	701	372	132	2	25
David	10/23/2007 11:21:20 AM	93	774	374	133	3	19

Show this minute

Figure 10.4 Seconds Out of Bounds

10.4 View Data Online

This interface shows the graphic that corresponds to the last 200 points that have arrived at the server. This data is refreshed every 200 milliseconds.

55
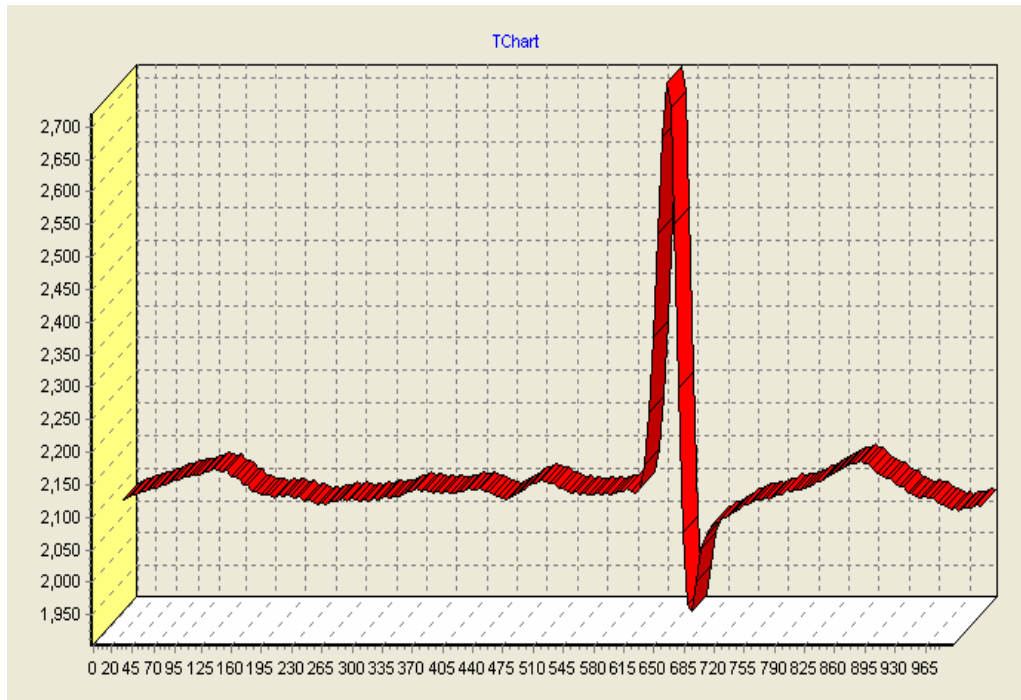
Figure 10.5 Online ECG

10.5 Prediction

In the prototype one example of data that can help to predict a disease has been implemented. This interface asks for a range of time and heart rate and looks in this range for times with a heart rate in this range and produces the graph representing the changes in amplitude for these instances over time. A continuous decrease of amplitude can predict a coronary heart disease.
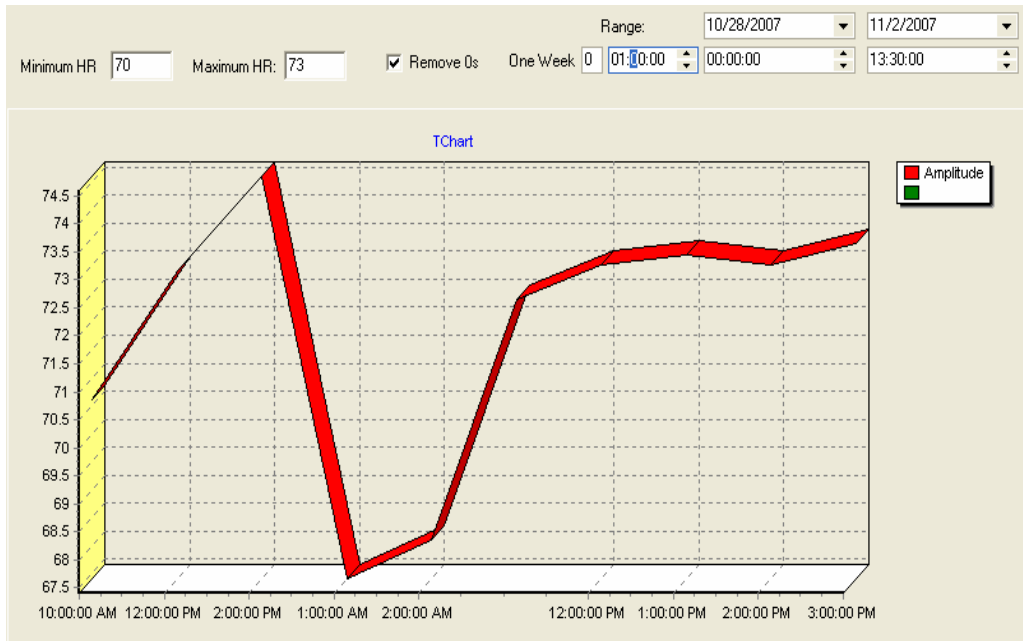
Figure 10.6 Predictions

CHAPTER 11

CONCLUSION AND FUTURE WORK

When first stage of the experiment was complete, connecting the ECG recorder and the PDA, we were surprised with the reliability of the ECG recorder; after the system was completed we discovered the weakness of the ECG. A new version of the ECG is being developed by the production company which might address some of these difficulties.

Despite the fact that the prototype system produced for the experiments was very limited, we were able to use it to obtain and compare data for different times and scenarios. Thanks to the database we could focus on similarities and differences in the data using a fixed range or one variable one. For example, we compared the behavior of the amplitude over the time when the patient had the same heart rate and found few changes but not continuous decrement. According to the literature[21], finding continuous decrement could signal a possible heart attack.

The system was tested inside a house and demonstrated how a person can live a normal life while he is monitored. A person uses the system during four days. Every day it was used between 4 and 8 hours covering all the normal activities of a person. Sleeping was the only activity that he did not realize. Also, the test showed that the system and the ECG recorder need some improvements when the patient practices

sports. The noise generated by the disconnections between the device and the skin is sometimes impossible to remove. It rendered around ten percent of the samples useless and sometimes produced a total disconnect of the system. For a remote monitor, this connectivity issue is a very important point to try to improve. The doctor should not be distracted by too many false positive alerts and the patient must not be scared by disconnects or malfunction alerts.

The integration of other devices can assist. For example, an Oxymeter could serve as a backup of the ECG, telling the system the heart rate, so the system can detect when the data from the ECG is incorrect.

We also detect a high necessity of at least an accelerometer or other device to help the system detects what the patient is doing. This is important because a detected acceleration of the heart rate in the system could be caused by the patient doing exercise or by an anomaly if he is resting.

We think this is only the beginning of a bigger project that will integrate medical researchers electronic engineers, doctors and computer people. This project will integrate more medical devices and new sensors that allow the system to detect other medical data. For example, sensors could detect when the patient took medicines, when the patient enters the kitchen or opens the freezer, if the patient is in the bed, what time he/she went to the bed, or when he/she woke up.

---

[21] Dean Jenkins,Stephen Gerred (2002) ECG Library Retrieved December 6 2007 from
http://www.ecglibrary.com/

With a system like this, the analysis of specific diseases like sleep apnea could be easier. The doctor can focus only on the data at the time the patient is asleep and only when the ECG shows abnormal data.

Also the system could be an invaluable tool for researchers due to the large amount of data that can be collected from different patients in different situations and the power of digital analysis using a database.

REFERENCES

1. Wan He, Manisha Sengupta, Victoria A Velkooff and Kimberly A de Barros (December 2005) 65+ in the United States: 2005. Retrived October 3 2007 from http://www.census.gov/prod/2006pubs/p23-209.pdf

2. CORSCIENCE (2007) CorBELT System. Retrieved December 6, 2007 from http://www.corscience.de/index.150.en.html

3. CORSCIENCE (2007) Bluetooth Blood Pressure Meter. Retrieved December 6, 2007 from http://www.corscience.de/index.158.en.html

4. CORSCIENCE (2007) Pulse Oximeter OxiPen. Retrieved December 6, 2007 from http://www.corscience.de/index.72.en.html

5. Wikipedia (2007) Thermoregulation. Retrieved December 6, 2007 from http://en.wikipedia.org/wiki/Thermoregulation

6. David C. Klonoff (May 10, 2007) Chapter 21. MONITORING TECHNOLOGIES- CONTINUOUS GLUCOSE MONITORING, BIOMARKERS OF GLYCEMIC CONTROL, ARTIFICIAL PANCREAS Retrieved December 6, 2007 from http://www.endotext.org/diabetes/diabetes12/diabetes21.html

7. CORSCIENCE (2007) Bluetooth Peak Flow Meter. Retrieved December 6, 2007 from http://www.corscience.de/index.154.en.html

8. Wikipedia (2007) Personal digital assistant Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Personal_digital_assistant

9. GSM Association (2007) GPRS Plataform Retrieved December 6 2007 from http://www.gsmworld.com/technology/gprs/index.shtml

10. American Heart Association (2007) Electrocardiogram (EKG or ECG) Retrieved December 6, 2007 from http://www.americanheart.org/presenter.jhtml?identifier=3005172

11. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

12. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #History

13. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

14. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #P_wave

15. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

16. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

17. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #ST_segment

18. Wikipedia (2007) Electrocardiogram Retrieved December 6 2007 from http://en.wikipedia.org/wiki/Electrocardiogram #T_wave

19. Lara Hopley and Jo van Schalkwyk. (2006/07/24) ECG scribbles. Retrived October 5 2007

20. CORSCIENCE (2007) CorBELT System. Retrieved December 6, 2007 from http://www.corscience.de/index.150.en.html

21. Dean Jenkins,Stephen Gerred (2002) ECG Library Retrieved December 6 2007 from http://www.ecglibrary.com/Wikipedia Project (2 November 2007)

22. Electrocardiogram. Retrieved November 4 2007 from http://en.wikipedia.org/wiki/Electrocardiogram

23. Dean Jenkins, Stephen Gerred (12th October 2002) ECG library. Retrieved November 4 2007 from http://www.ecglibrary.com/

24. Richard E. Klabunde (April 6 2007) Cardiovascular Physiology Concepts. Retrieved November 4 2007 from http://cvphysiology.com/

25. CORSCIENCE (2007) CORSCIENCE. Retrieved November 4 2007. from http://www.corscience.de/en-index-html.2.en.html

26. CORSCIENCE (2007) 12-channel ECG Device with Bluetooth Functionality BT12. Retrieved November 2007 from http://www.corescience.de/12-channel-ecg-device-with-bluetooth.37.en.html

27. Bluetooth Sig Inc (2007) Bluetooth technology. Retrieved November 4 2007 from http://www.bluetooth.com/Bluetooth/Learn/Technology/

28. Microsoft Corp (2007) What's New in Visual Studio 2005 for Native Developers Retrieved November 4 2007 from http://msdn2.microsoft.com/en-us/library/ms838270.aspx

29. Girish NS (Octuber 23 2007) Introduction to Pocket PC Development. Retrieved November 4 2007 from

    http://www.codeproject.com/ce/Pocket_PC_Development.asp

30. Practical Guide for Programmers (ISBN: 1-55860-826-5) by Michael J. Donahoo and Kenneth L. Calvert

31. Microsoft Corp (2007) SQL Server 2005 Express Edition Retrieved November 4 2007 from http://msdn2.microsoft.com/en-us/sql/Aa336346.aspx

32. SparkFun (2005) Sensors/ Accelerometers Retrieved November 4 2007 from

    http://www.sparkfun.com/commerce/product_info.php?products_id=254

33. Noin Medical Inc (2007) Bluetooth-Enabled Pulse Oximeter. Retrieved November 4 2007 from http://www.rtx.dk/Nonin_Medical,_Inc-802.aspx

34. Engineeringtalk (June 4 2004) Bluetooth enables wireless

    blood pressure monitor. Retrieved November 4 2007 from

    http://www.engineeringtalk.com/news/pym/pym102.html

35. Wikipedia (2007) SQL. Retrieved November 4 2007 from

    http://en.wikipedia.org/wiki/SQL

36. Coordinated Care Solutions (2004) Careguide@home. Retrieved November 4 2007 from http://www.eldercare.com/

37. Senior Solutions of America (2007) Aging parents and elder care. Retrieved November 4 2007 from http://www.aging-parents-and-elder-care.com/

38. American Sleep Apnea Association (2007) SLEEP APNEA INFORMATION. Retrieved November 4 2007 from http://www.sleepapnea.org/info/index.html

39. Wikipedia Org (2007) Myocardial infarction. Retrieved November 4 2007 from http://en.wikipedia.org/wiki/Myocardial_infarction

BIOGRAPHICAL INFORMATION

Jorge David Herrera was born in Colombia. After graduating from Universidad de los Andes in Bogotá Colombia, he was hired by this University to teach and administrate the engineering computer labs. One and a half year later, he was hired by San Martin University to create and administrate the faculty of computer science. One and a half years later he had to leave this job due the impressive success of his company SOLINFO LTDA and his main product COMPUCILINA.

He won the most important award for a computer science engineer in Colombia in 1994, the Colombian Informatics award, brought by the Colombian society of computer science engineers.

Jorge's research interest includes software development best practices, quality assurance and low level programming. Jorge's company LEVERIT will introduce new products to the US market next year.