HEARTFAID

D14 – Specifications of Data acquisition and transmission infrastructure

Submission date: 09/03/2007
Due date of document: 31/01/2007

Information Society and Media

HEARTFAID

A KNOWLEDGE BASED PLATFORM OF SERVICES FOR SUPPORTING MEDICAL-CLINICAL MANAGEMENT OF THE HEART FAILURE WITHIN THE ELDERLY POPULATION

Project summary	
Project acronym:	HEARTFAID
Project identifier:	IST – 2005 – 027107
Duration of the Project:	01/02/2006 – 31/01/2009
Project Co-ordinator:	UNICAL University of Calabria (Italy)
Thematic Priority:	Information Society Technology
Instrument:	Specific Targeted Research or Innovation Project

Consortium
➢ UNICAL - Università della Calabria (Italy)
➢ UNICZ - Università degli studi Magna Graecia di Catanzaro (Italy)
➢ UNIMIB - Università degli studi di Milano Bicocca (Italy)
➢ JUMC - Jagiellonian University Medical College (Poland)
➢ VMWS - Virtual Medical World Solutions Ltd (United Kingdom)
➢ FORTHNET S. A.- Hellenic Telecommunications and Telematic Applications Company S. A. (Greece)
➢ SYNAP - Synapsis s.r.l. (Italy)
➢ CNR - Consiglio Nazionale delle Ricerche (Italy)
➢ FORTH - Foundation for Research and Technology Hellas (Greece)
➢ RBI - Rudjer Boskovic Institute (Croatia)
➢ AUXOL - Istituto Auxologico Italiano (Italy)

D14 – Specifications of Data acquisition and transmission infrastructure

Document summary	
Document title:	D14 – Specifications of Data acquisition and transmission infrastructure
Document classification:	Derivable D14
Dissemination level:	CO
Submission date:	09 March 2007
Due date:	31 January 2007
Authors:	Christos Biniaris – VMWS Marianna Dikaiakou – Forthnet Manolis Stratakis - Forthnet Stelios Louloudakis – Forthnet Sergio Di Bona - SYNAP
Work package:	WP2 – BIOMEDICAL DATA IDENTIFICATION AND COLLECTION
Report version:	2.0

Short description
The goal of this deliverable is to define requirements and specifications related to the data acquisition and transmission infrastructure, which involves all the modules required to collect the biomedical data from the sensors and transmit these data into the HEARTFAID platform.

Change record		
Version number	Changes	Release date
0.1	Template	16/01/2007
0.2	Preliminary draft of the document	08/02/2007
0.3	First Draft of the document	15/02/2007
1.0	Advanced Draft	23/02/2007
1.5	Final Draft	05/03/2007
2.0	Final Version	08/03/2007

Table of contents

Executive Summary

The main objective of the HEARTFAID project is to devise, develop and validate an advanced and innovative technological platform of services and end-user applications aiming to contribute towards the optimization of the clinical management of HF and the reduction of the economic and social costs, by collecting, integrating and processing all types of the above mentioned biomedical data and information. In particular, the early detection of HF related signs and symptoms and the appropriate identification and acquisition of biomedical data from myocardial tissue and organ, may contribute to delay the hospitalization and to improve both the quality of life and survival in pathologic patients.

The identification and acquisition of the relevant biomedical data in the scope of the HEARTFAID project takes place in WP2 – Biomedical Data Identification and Collection and Task 2.2 of this WP is related to the Design and Development of the Data Acquisition and Transmission Infrastructure, which will enable the biomedical data acquisition as well as the transmission of these data to the HEARTFAID platform.

The biomedical data Acquisition and Transmission infrastructure, therefore, provides the required mechanisms in order

- to acquire the biomedical data from the relevant sensors in all the identified healthcare environments and
- to transmit these data to the HEARTFAID platform

The Data Acquisition and Transmission Infrastructure comprises a set of elements in order to fulfill the data acquisition and transmission process. These elements range from sensors which will perform the measurements, to communication modules in order to communicate the measurements, devices to collect the data, software modules, APIs, data tansmission modules and data storage mechanisms.

The purpose of the current Deliverable is to report the requirements and specifications related to the aforementioned components and finally conclude to an architecture, on which the first prototype of the Data Acquisition and Transmission Infrastructure will be based.

More specifically, the organisation of the document is the following:
At first, an introduction will provide a general overview of the field and afterwards, the general objectives related to the overall functionality of the Data Acquisition and Transmission Infrastructure are given and the methodological approach towards turning these objectives into requirements and specifications is presented.

After the methodological approach is given, a detailed technological background is presented aiming to illustrate what is the current status in the field and what type of functionality we can expect from the Data Acquisition and Transmission infrastructure in the scope of HEARTFAID.

After having presented the technological background in detail, the system requirements are presented. More specifically, some general requirements are given in the beginning, and next the healthcare environments are identified and the requirements related to each one of these environments are presented.

Having defined the system's requirements, the building blocks of the system are defined and the requirements related to each one of them are presented in more detail.

Next, the specifications of the system's components are illustrated starting from the presentation of the overall architecture of the data acquisition and transmission infrastructure and continuing with more detailed specifications related to the various components of the infrastructure.

The document ends with a conclusion section aiming to recapitulate all the information presented in the document and link the outcome of the document to future activities towards the implementation of the first prototype of the Data Acquisition and Transmission infrastructure.

1 Glossary of terms

TERM	DEFINITION
ACL	Asynchronous Communication Link
ADSL	Asymmetric Digital Subscriber Line
AFH	Adaptive Frequency Hopping
AMR	Automated medical record
ANSI	American National Standards Institute
APIs	Application Programming Interfaces
ARQ	Automatic Retransmit Request
ASIC	Application-specific integrated Circuit
ASK	Amplitude Shift Key modulation
ASP	Active Server Pages
AWT	Abstract Window Toolkit
Bluetooth SIG	Bluetooth Special Interest Group
BPS	Bits per second
BPSK	Biphase Shift Key Modulation
CDMA	Code division multiple access
CLDC	Connected Limited Device Configuration
CPR	Computerised medical record
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
CSMA/CD	Carrier sense multiple access with collision detection
DCE	Data Communications Equipment
DID	Bluetooth Device ID Profile
DOM	Document Object Model
DSL	Digital subscriber line
DSSS	Direct Sequence Spread Spectrum
DTE	Data Terminal Equipment
EDR	Enhanced Data Rate
EEG	Electroencephalogram
EHCI	Extended Host Controller Interface
EHR	Electronic Health Record
EKG (or ECG)	Electrocardiogram
EMR	Enterprise electronic medical record
EOG	Electroculogram
EPR	Electronic patient record
ER	Electronic Record
eSCO	Extended Synchronous Connections
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FHSS	Frequency Hopping Spread Spectrum
FSK	Frequency shift key modulation

FTP	File Transfer Protocol
GP	General Practitioner
GPL	General Public License
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HCD	Host Controller Device
HDSL	High Data Rate Digital Subscriber Line
HF	Heart Failure
HIPAA	Health Insurance Portability and Accountability Act
HIS	Health Information System
HTTP	Hypertext Transfer Protocol
I/O	Input / Output
IC	Integrated Circuit
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IrDA	Infrared Data Association
ISI	Intersymbol Interference
ISM	Industrial, Scientific, and Medical Band
J2ME	Java2 Micro Edition
J2SE	Java2 Standard Edition
JAD	Java Application Descriptor File
JAXP	Java API for XML Processing
JCA	Java Cryptography Architecture
JCE	Java Cryptography Extension
JSR	Java Specification Request
JVM	Java Virtual Machine
L2CAP	Bluetooth Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LED	Light Emitting Diode
MAC	Message Authentication Code
MICS	Medical Implant Communications Service Band
MIDP	Mobile Information Device Profile
MMS	Multimedia Messaging Service
OBEX	Bluetooth Object Exchange
OOP	Object Oriented Programming
OOSK	On-off shift key modulation
OS	Operating System
OSI	Open Systems Interconnection Model
PAN	Personal Area Network
PC	Personal Computer
PDA	Portable Digital Assistant
PHI	Protected Health Information

PKI	Public Key Infrastructure
POCCIC	Point-of- Care Connectivity Industry Consortium
POSIX	Portable Operating System Interface Standard
PSK	Phase Shift Key modulation
RADSL	Rate-Adaptive Digital Subscriber Line
RAM	Random Access Memory
RF	Radio Frequency
RFCOMM	Radio Frequency Communication
RFID	Radio Frequency Identification
ROM	Read Only Memory
RSA	Rivest, Shamir, and Adleman Encryption
RSSI	Received Signal Strength Indication
SAFER	Secure And Fast Encryption Routine
SAX	Simple API for XML Parsing
SCO	Synchronous Connection Oriented link
SDAP	Bluetooth Service Discovery Application Profile
SDK	Software Development Kit
SDSL	Symmetric Digital Subscriber Line
SMS	Short Messaging Service
SPP	Bluetooth Serial Port Profile
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
UART	Universal asynchronous receiver/transmitter
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
USB	Universal Serial Bus
VDSL	Very High Speed Digital subscriber line
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WAN	Wide Area network
WAP	Wireless Application Protocol
WMTS	Wireless Medical Telemetry System
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
XML	Extensive Markup Language
XNI	Xerces Native Interface
XSLT	Extensible Stylesheet Language Transformations
2.5G	enhanced Second Generation Mobile Networks
2G	Second Generation Mobile Networks
3DES	Triple Data Encryption Standard
3G	Third Generation Mobile Networks

2 Introduction

As stated in the executive summary, the purpose of this document is to define the requirements and the functional specifications of the Data Acquisition and Transmission Infrastructure in the scope of the HEARTFAID project. This infrastructure will facilitate the collection of biomedical data from the HEARTFAID patients in all the identified healthcare environments and subsequently the transmission of these data to the HEARTFAID platform.

In order to achieve these goals, a synergy of devices, components, sub-systems and technologies is required. The functionality implemented by this synergy shall be able to coexist with already existing infrastructures and extend their functionality in order to facilitate the objectives of the HEARTFAID project.

Considering clinical environments, sensors and devices already available in these environments have to be able to provide data into the Data Acquisition and Transmission Infrastructure; hence, the communication interfaces of the infrastructure have to be able to cooperate with these systems (usually providing serial interfaces for the communication with other devices).

However, since the major goal of HEARTFAID is the delay the hospitalization and the improvement of both the quality of life and survival in pathologic patients, it is implied that the technological "centre of gravity" related to the Data Acquisition and Transmission Infrastructure falls in the patient's environment, whether it is a home or an outdoor environment. Moreover, since the main objective is the ergonomic nature of the data acquisition infrastructures for these kinds of environments, it is evident that wireless sensor networks are utilised in great extent in this field.

The scenario of wireless sensor networks feeding information into the system brings in many parameters which need to be analysed when designing and implementing such an infrastructure. Characteristic parameters are, for instance, radio requirements, interference, communication protocols, data security and integrity etc.

Furthermore, considering the fact that, in future medical applications, the data acquisition takes place at the user's environment, the desirable solution for data acquisition is to use devices which exist in the user's environment in order to collect all the biomedical data in this environment. Solutions of this type can be for example an internet connected PC in the user's home or the mobile phone of the user in an outdoor environment. In order to examine the way that these devices can be used for data acquisition, one has to survey their communication capabilities (both to the sensors and to the HEARTFAID platform via public networks), the features offered by their operating systems, their programmability and the available APIs (e.g. for communication, data processing, data encryption etc).

Finally after the data is sent to the HEARTFAID platform, one has to consider the way this data is stored and made available to other entities of the HEARTFAID platform.

Considering the above, the main macroscopic building elements of the data acquisition and transmission infrastructure are:

i) the medical sensors
ii) the communication mechanisms both to the sensors and to the HEARTFAID platform
iii) the devices that collect the data from the sensors
iv) the data storage mechanisms

Taking into account these building elements, and in order to define the functional specifications for the modules which constitute the Data Acquisition and Transmission infrastructure, we adopt the following logical procedure which is reflected to the structure of the document:

In Section 3 the general objectives of the data acquisition and transmission infrastructure are defined, while in Section 4 the methodological approach towards the definition of the specifications of the system's components is presented. Next, in Section 5 a detailed technological background sets the technological basis of the infrastructure, while Section 6 defines the requirements relative to the functionality of the infrastructure, leading to a more detailed definition of the infrastructure's elements as well as their roles and their more detailed requirements. Finally, in Section 7 an overall architecture is illustrated and the specification of its elements is given.

3 General Objectives

As described in the TA, the main objectives of the data acquisition and transmission infrastructure are:

a) ***to acquire*** all the relevant biomedical data within the several healthcare environments
b) ***to transmit*** the acquired data to the middleware of the HEARTFAID platform

Towards meeting these objectives, we can identify some high level criteria which will assist towards the definition of the infrastructure's overall requirements. We could say that the Data Acquisition and Transmission Infrastructure should be ideally able to easily integrate any new sensor, while also it should be compatible with the already existing systems. Furthermore, it should use already existing equipment and be based on easily affordable components and technologies.
In order to achieve such functionality, the design of such an infrastructure should be based on the following principles:

- use of ***standardised and widely accepted technologies*** in order to build the elements of the infrastructure
- adopt ***a modular design of the infrastructure*** in order, not only to be able to develop and test each part independently, but also to be able to cope with future technological advances dictating the replacement of a specific module by a new one.

It is obvious the general objectives regarding the functionality of the Data Acquisition and Transmission infrastructure together with the detailed analysis of the related technological background will contribute towards the definition of the building elements of the infrastructure and the identification of the roles of these elements.

4 Methodological Approach

Having defined the general objectives of the data acquisition and transmission infrastructure, the next logical step is to define the methodology according to which this infrastructure will be designed and developed.
The first step is the **identification of the technological background** and state of the art. Since the design and development of such an infrastructure involves a variety of components and modules including sensors, communication modules, communication protocols, computers, mobile devices, software modules etc, the available technologies related to each one of them need to be surveyed in order to clearly identify the system's functionality also have a clear view on what options are available in order to achieve to desired goals.

Based on the definition of the desired functionality, the next step is the **definition of the requirements**, i.e. the identification what is really needed in order to fulfill this functionality. The first step towards the definition of requirements is the identification of each healthcare environment examined in HEARTFAID. In this scope the involved entities and the requirements related to each of these environments are identified.

Having defined the requirements of the system, **the system's components are identified**, as well as their roles in the infrastructure are defined. By having defined the components and their roles, one can progress one step further and define more detailed requirements imposed by the use of these components.

Finally, having defined the system's components, an overall architecture of the Data Acquisition and Transmission Infrastructure is presented and finally the detailed **specifications of these components**, i.e. the characteristics of these components are identified.

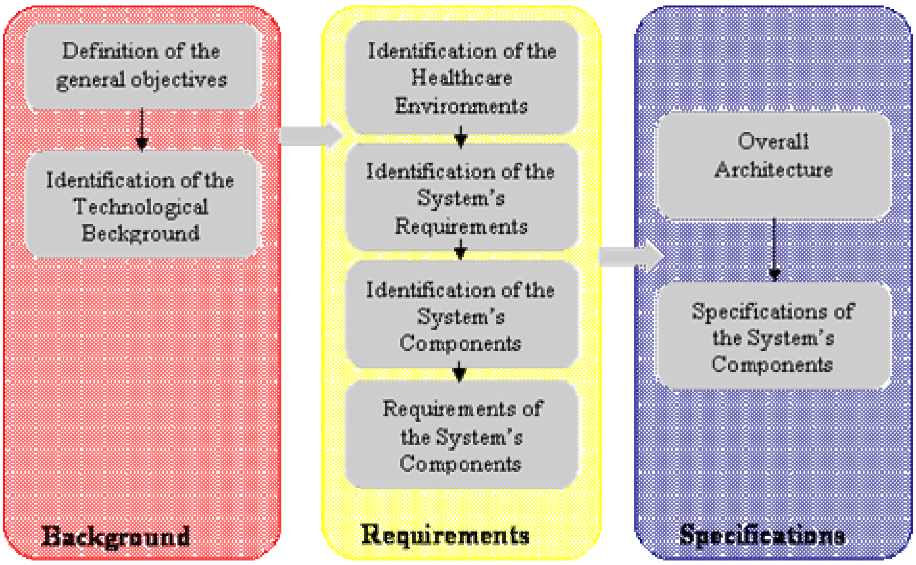


Figure 1 Methodological Approach

5 Technological Background

In this section, the technological background related to the design and development of the Data Acquisition and Transmission infrastructure is presented. The section begins with the description of future medical applications, which revolve around the concept of homecare and selfcare in the user's environment, together with the technological advance in wireless sensor networks and body area networks. In this context, several key technologies and applications are presented in more detail (e.g. Bluetooth Applications in Medical field). Regarding the data transmission process, transmission technologies involving public fixed and wireless networks are illustrated. Finally, technologies related to the storage of biomedical data are presented.

5.1 Medical Applications of the future

Advances in wireless sensor networking have opened up new opportunities in healthcare systems. The future will see the integration of the abundance of existing specialized medical technology with pervasive, wireless networks. They will co-exist with the installed infrastructure, augmenting data collection and real-time response. Examples of areas in which future medical systems can benefit the most from wireless sensor networks are in-home assistance, smart nursing homes, and clinical trial and research augmentation.

As the world's population ages, those suffering from diseases of the elderly will increase. In-home pervasive networks may assist residents by providing memory enhancement, control of home appliances, medical data lookup, and emergency communication.

Unobtrusive, wearable sensors will allow vast amounts of data to be collected and mined for next-generation clinical trials. Data will be collected and reported automatically, reducing the cost and inconvenience of regular visits to the physician. Therefore, many more study participants may be enrolled, benefiting biological, pharmaceutical, and medical-applications research.

5.2 Implementation issues related to Future Medical Devices

The following list describes in more detail some kay implementation issues related to the future medical devices.

- *Interoperability*: As a result of the current heterogeneity, communication between devices may occupy multiple bands and use different protocols. For example, unlicensed bands are many times used for general telemetry or ISM equipment. Implanted medical devices may use a licensed band allocated for that purpose by the FCC. In order to avoid interference in the increasingly crowded unlicensed ISM band, biomedical devices may use the WMTS band (wireless medical telemetry services, at 608 MHz). ***The homecare network must provide middleware interoperability between disparate devices, and support unique relationships among devices***, such as implants and their outside controllers.

- *Real-time data acquisition and analysis***:** The rate of collection of data is higher in this type of networks than in many environmental studies. ***Efficient communication and processing*** will be essential. Event ordering, time-stamping, synchronization, and quick response in emergency situations will all be required.
- *Reliability and robustness*: Sensors and other devices must operate with enough reliability to yield ***high-confidence data suitable for medical diagnosis and treatment***. Since the network will not be maintained in a controlled environment, devices must be robust.
- *New node architectures*: The integration of different types of sensors, RFID tags, and back-channel long-haul networks may necessitate ***new and modular node architectures***.

5.3 Benefits from the use of wireless sensors

When a system is wired, the wires limit the number of components that can be plugged together, the distances between the components, and the number of layers of clothing that can be crossed. Without wires, these constraints disappear and many benefits come along:

- ***Size and location***. When the number of components increases, the responsibility of each decreases. Instead of a one-size-fits-all set of components, each component can tailor itself to one particular task and optimize its shape, size, and location accordingly. For example, medical monitoring components such as heart monitors can be located close to the appropriate area of the body without "dragging" the rest of the system with them, and the shape need only accommodate the monitoring function.
- ***Customization***. Because components are specialized and removable, they can also be tailored—users can hot swap user interface components to fit their particular needs or preferences. For example, a user's WPAN might have several different feedback modules (tactile, visual, or auditory) for different contexts. The user can tailor each of those modules to the user's preferences.
- ***Consolidation***. Components can be easily shared among applications. For example, there is no reason for both a cell phone and pager to contain a list of phone numbers—they should consolidate their information. I/O devices such as headphones, displays, speakers, and microphones could serve any application that needs them.
- ***Kinesthetics*.** Consumers generally prefer wireless devices because wires can tangle, restrict movement, be tripped over, and get caught on other objects. Devices such as WPAN-participating wristwatches would most likely not be accepted commercially if wired to other wearables.
- ***Unobtrusiveness***. Making the system smaller, wireless, and less obvious to other people lessens the "cyborg look" and makes the system less obtrusive. So, the perceived distancing effect of having a wearable system would not come into play as often or as deeply.

- ***Multiplicity***. Because we can have many more components, we can readily support multiples of any particular component (for example, multiple user interface components). Because the components don't use physical connectors, they can simultaneously interact.
- ***Transferability***. When the components become multiple and wireless, users can capriciously pick them up, set them down, or hand them off, making them more transferable.

All these benefits can be synergistic and their combinations provide new artifacts and use scenarios. However, compared to their wired counterparts, WPANs do have some disadvantages, the first of which is ***less available power***. Small devices are constrained by their battery size, and practical wireless transfer of power between components is infeasible. Many researchers are investigating ways to reduce power consumption and improve power generation. There is also an association problem. If two PAN components are near each other, are they part of the same PAN? In a wired system, the answer is simple—if they are connected to each other, they are. In a wireless system, the answer is no longer clear—with which PAN is a component associated? This is an active research area, with several solutions under investigation.

Another disadvantage is that ***data rates are reduced***. Maximum data rates for wires far outstrip wireless connections. Many applications, however, have modest data rate requirements.

Finally, a ***wireless protocol standard is needed***. Wireless protocols have many more constraints than their wired counterparts with respect to range, frequency spectrum, and so on. The Bluetooth wireless communication protocol has emerged to address this. It is ideal for WPANs because its nominal range is only 10 meters. It consumes less power and has much less range than other standards that target wireless LANs such as 802.11. Bluetooth is thought to be available in over 70 percent of mobile handsets, today. Bluetooth's lower layers have been adopted as the IEEE 802.15.1 standard.

So, although there are disadvantages, they are tractable and diminishing, and the benefits outweigh the negatives.

5.4 Characteristic Example: Bluetooth technology in the Medical field

Like commercial applications, medical devices also face many development issues that can be addressed with wireless technologies. Bluetooth is particularly well suited for cable replacement, allowing for mobile connectivity. It also provides excellent security and reliability and coexists well with other wireless technologies. And, it is a relatively low-cost technology to implement. Several medical systems have been implemented successfully. The major characteristics of offered by Bluetooth technology, are illustrated in the following lines:

- **Cable Replacement**: Bluetooth was designed for cable replacement. It replaces relatively low-data-rate connections such as the traditional RS-232 used in many medical devices with data rates in the 9600–115,200-baud range. It provides a wireless connection and eliminates the need for expensive isolation and potentially hazardous cabling.
- **Connectivity**: Using IEEE standardized technology (802.15.1), Bluetooth allows connection with off-the-shelf components and enables connectivity not only to personal-area networks (PANs), but also to local-area networks (LANs) and wide-area networks (WANs) through access points and cellular handsets. By using a standard interface, issues associated with incorporating quickly changing technology into a medical device are no longer a concern.
- **Reduced Power Consumption**: Bluetooth offers considerable power savings compared with other wireless standards that are also designed for highly mobile, battery-powered devices. Power consumption is only 10–25% of that of 802.11b. For devices that do not continuously transmit data, Bluetooth provides several power-saving modes to decrease power further.
- **Security**: With 128-bit SAFER+ encryption and authentication, the technology provides security. Controllable discovery and connectability modes control access. This is particularly important for safe connections needed for mobile data devices in the medical environment. The SAFER+ encryption has yet to be broken; however, successful attacks on devices with incorrectly implemented security policies have been reported. The security provided meets the requirements for medical device and patient information data.
- **Reliability**: The frequency-hopping, spread-spectrum technology is highly tolerant of ambient radio-frequency (RF) noise and retains good bandwidth even in the presence of devices such as electrosurgical units and home appliances.
- **Coexistence**: Testing performed to date shows low levels of interference with other technologies, including 802.11b, in the ISM band.

5.5 Existing Bluetooth Medical Applications

Some of the existing bluetooth medical applications, which demonstrate the breadth of capabilities and solutions that can be achieved in medical devices with Bluetooth wireless technology are highlighted below.

- **Stryker Endoscopy Sidne**. The Stryker system is the first in-hospital system incorporating Bluetooth to receive 510(k) clearance. Bluetooth is used to eliminate cabling from the controller unit to devices at the operating table. It eliminates cabling issues caused by debugging and replacement, which can lead to downtime of the operating room. The Sidne application uses 802.11b technology as well as Bluetooth for high-bandwidth data (images).

- **Zoll M-Series Defibrillator/Monitor**. The Bluetooth accessory in this device enables transmission of 12-lead and vital-signs trend reports to a PDA. It eliminates memory cards and enables emergency services personnel to remotely transfer data and reports. This device is used in a battery-constrained environment where cabling is not only a nuisance, but also presents a hazard to patients and caregivers.
- **GE Medical Achilles DexterQUS**. This device involves measurements taken at floor-level on a patient's foot or heel. Bluetooth allows the GE system to be used at the patient's level instead of needing to be on the floor. The PDA software facilitates data entry and report printing without wires. The wireless interface also allows for data to be downloaded to a PC for data archiving and future retrieval. This enables caregiver mobility using an off-the-shelf PDA and eliminates bothersome and hazardous cabling.
- **Nonin Pulse Oximeter**. The Nonin system eliminates the need for a wire between the patient and a portable monitor, which allows patient mobility within Bluetooth's 10-m radius. The battery life for continuous monitoring exceeds 4 days, eliminating frequent charging or battery changes. The Nonin system exemplifies how Bluetooth can enable highly mobile sensor-based applications. It has to be noted that Nonin Pulse Oximeter is among the bluetooth medical devices that will be integrated in the HEARTFAID platform
- **RTX Healthcare Wireless TeleHealth**.The RTX components facilitate systems for monitoring the weight and blood pressure of patients in their homes. The measurement devices interface to a home gateway allowing remote downloading of collected data. Mobility and interfacing to industry standards are key elements of the system.

5.6 Embedded, Real-Time, Networked System Infrastructures

The state of the art embedded, real-time networked system infrastructures involve the following issues:

- *Patient and object tracking:* Tracking can be considered at three levels: symbolic (e.g., Room 136 or X-Ray Lab); geographical (GPS coordinates of a patient on an assisted living campus); relational/associational ("Dr. Marvin is currently with Patient Bob"). It is complicated by the presence of multiple patients, non-patient family members, and leaving the range of the home network.
- *Communication amid obstructions and interference*: In-building operation has more multi-path interference due to walls and other obstructions, breaking down the correlation between distance and connectivity even further. Unwanted emissions and glitching are likely to be rigorously restricted and even monitored due to safety concerns, particularly around traditional life-critical medical equipment.
- *Multi-modal collaboration and energy conservation*: Limited computational and radio communication capabilities require collaborative

algorithms with energy-aware communication. Richly varied data will need to be correlated, mined, and altered. Heterogeneous devices will be on very different duty-cycles, from always-on wired-power units to tiny, stealthy, wearable units, making rendezvous for communication more difficult.

- *Multi-tiered data management:* Data may be aggregated and mined at multiple levels, from simple on-body filtering to cross-correlation and history compression in network storage nodes. Embedded real-time databases store data of interest and allow providers to query them.

5.7 Technologies for the transmission of the acquired medical data

These technologies will enable the transmission of the biomedical data to the HEARTFAID platform. Generally, we can identify two major types of transmission, the wireless and the fixed. Relevant technologies which will enable data transmission are presented in the following sub-sections, focusing on the medical field.

5.7.1 Public wireless networks supporting health services

The introduction of third generation (3G) public wireless network infrastructures, such as the Universal Mobile Telecommunications System (UMTS), enables the development of innovative mobile services. For example, deploying m-health services which embed tele-monitoring and tele-treatment services become feasible with the role-out of 3G networks. These services allow healthcare professionals to monitor a mobile patient's vital signs and provide feedback to this patient anywhere and any time. The performance of m-health services perceived by end-users depends on the serviceableness of 3G networks to support these services. Hence, the performance of 3G networks is a critical factor for successful development of m-health services.

The emerging high bandwidth mobile operator networks (e.g. 3G networks) combined with the ever-advancing miniaturization of sensor devices and computers will give rise to new mobile services and applications that will affect and change the daily life of citizens. An area where these new technological advances will have a major effect is healthcare. Citizens, being patients or non-patients, will not only be able to get medical advice from a distance but will also be able to send from any location full, detailed and accurate vital signal measurements, as if they had been taken in a medical center, implementing what we can call "ubiquitous medical care".

Successful tele-monitoring service delivery strongly depends on the ability of the transport system to support the service requirements. For example, for healthcare professionals to use the monitoring service on chronic patients successfully, it is very important to disclose and understand the transport system's behaviour. For example, the supported service data type (i.e. vital signs data, text, audio, and video) and data volume strongly depend on the capacity (e.g. bandwidth) of the underlying transport system.

5.7.2 Broadband Fixed Networks used in Clinical and Home Environments

Wireless networks are gaining popularity, due to their omnipresent availability, but fixed networks remain the most robust and widespread connectivity option in medical and other time-critical systems. Where a fixed network is available, its use is preferred over a wireless one, because of the enhanced security, greater bandwidth and lower cost. HEARTFAID aspires to make medical care available in a variety of environments, where a fixed network is in fact available and can be utilized. Such environments are the Home and the Clinical, where both expert and non-expert users are expected to interact with the HEARTFAID Front-End. The networking is very different for each of the two cases, and is presented in detail below.

- **Home Environment**

A broadband connection to the Internet can be used at the patients at home, in order to provide a prompt transmission of the collected sensor data from their own settings to the central HEARTFAID middleware platform. The common broadband option for today's users is the Digital Subscriber Line (DSL), a technology that provides high-speed data transmissions over the so-called "last-mile" of "local-loop" of the telephone network. DSL technologies can enhance copper wire infrastructure to be the most effective way of delivering broadband services to the greatest number of people

DSL connections are point-to-point dedicated circuits, meaning that they are always connected. There is no dial-up. There is also no switching, which means that the line is a direct connection into the carrier's system. DSL modems are required at the home site and the carrier site.

DSL comes in seven different flavors, in speeds ranging from 16 Kbits/sec to 52 Mbits/sec. The services are either symmetric (traffic flows at the same speed in both directions) or asymmetric (the downstream capacity is higher than the upstream capacity). A description of the different version follows. These versions are often collectively referred to as xDSL.

- **HDSL (High-Speed Digital Subscriber Line)** HDSL is the most common and mature of the DSL services. It delivers data symmetrically at T1 data rates of 1.544 Mbits/sec over lines that are up to 3.6 kilometers (12,000 feet) in length. Generally, HDSL is a T1 service that requires no repeaters but does use two lines. Voice telephone services cannot operate on the same lines. It is not intended for home users, but instead is intended for the telephone company's own feeder lines, inter-exchange connections, Internet servers, and private data networks.

- **SDSL (Symmetric Digital Subscriber Line)** SDSL is a symmetric bidirectional DSL service that is basically the same as HDSL, but operates on one twisted-pair wire. It can provide data rates up to the T1 rate of 1.544 Mbits/sec.

- **ADSL (Asymmetric Digital Subscriber Line)** ADSL is an asymmetric technology, meaning that the downstream data rate is much higher than the upstream data rate. As mentioned, this works well for a typical Internet session in which more information is downloaded from Web servers than is uploaded. ADSL operates in a frequency range that is above the frequency range of voice services, so the same wire can carry both analog voice and digital data transmissions. The upstream rates range from 16 Kbits/sec to as high as 768 Kbits/sec.

- **VDSL (Very High-Data-Rate Digital Subscriber Line)** VDSL is basically ADSL at much higher data rates. It is asymmetric and, thus, has a higher downstream rate than upstream rate. The upstream rates are from 1.5 Mbits/sec to 2.3 Mbits/sec. VDSL is seen as a way to provide very high-speed access for streaming video, combined data and video, video-conferencing, data distribution in campus environments, and the support of multiple connections within apartment buildings.

- **RADSL (Rate-Adaptive Digital Subscriber Line)** This service is also similar to ADSL, but it has a rate-adaptive feature that will adjust the transmission speed to match the quality of the line and the length of the line. A line-polling technique is used to establish a connection speed when the line is first established.

- **DSL Lite (or G.Lite)** DSL Lite is considered a "jump-start" technology that is meant to deliver DSL to the greatest number of users, as fast as possible. While it has a lower data rate than other DSLs, it does not require that the telephone company do anything to the lines. In addition, equipment to handle DSL Lite is becoming readily available at a low price.

From the analysis above, considering the monitoring of patients at their home environment, the decision for the most appropriate DSL version to be used should be related to the need for a fast upload data stream. This consists in the demand for fast transmission of the data, which is acquired from the medical sensors, to the central middleware of the platform. Furthermore, the availability of each of the DSL versions described above at the patient's house should also be considered, since there are some implementation requirements for each version.

- **Clinical Environment**

Network integration and interconnection of medical devices in a Hospital is a challenging and important undertaking. An increasing number of modern hospitals take advantage of the automation and robustness offered by networked settings, in order to provide higher quality services to every patient. An invaluable

tool for fast and reliable LAN connectivity, such as that of a Hospital, is the Ethernet standard, which was established in 1980 under the IEEE 802.3 set of specifications.
Traditional Ethernet supports data transfers at the rate of 10 Megabits per second (Mbps). Over time, as the performance needs of LANs increased, the industry created additional Ethernet specifications for Fast Ethernet and Gigabit Ethernet. Fast Ethernet extends traditional Ethernet performance up to 100 Mbps and Gigabit Ethernet up to 1000 Mbps speeds. Although products aren't yet available to the average consumer, 10 Gigabit Ethernet (10000 Mbps) also remains an active area of research.
The Ethernet’s technical aspects have evolved over time to reach the level of maturity that is encountered in the technology today.
Despite the huge changes in Ethernet from a thick coaxial cable bus running at 10 Mbit/s to point-to-point links running at 1 Gbit/s and beyond, all generations of Ethernet (excluding very early experimental versions) share the same frame formats (and hence the same interface for higher layers) and can be readily (and in most cases cheaply) interconnected.

5.7.3 Technologies for the data storage (Electronic Health Records)

Digital equipment and computerized systems are progressively invading all sectors of healthcare. The introduction of computer technology in healthcare been continuing for more than 30 years and has slowly influenced the way healthcare is provided. The progressive introduction of digital equipment in hospitals, reducing exploitation costs, increasing efficiency and optimizing processes (also from the environmental point of view reducing the waist of paper and minimizing the development of photographic films) has been clear on the last years and is now irreversible. This new technology motivated the creation of electronic health records (EHR) that can contain all the information about the patient including analysis, images and other tests.

Nevertheless, although the last years have seen an unprecedented increase on the digitalization of healthcare services, an important part of healthcare is still paper based using traditional methodologies. The increasing use of digital equipment has inspired many new studies that take advantage of the large databases nowadays available for research proposes have been developing new algorithms and methods for disease diagnosis and prediction. But only a low percentage of the healthcare sector is already in the digital era. If we pick the health record of a patient, we cannot expect to have much of it in a digital format. Though the situation clearly depends upon the country and the region of the country, even in the most developed institutions we cannot expect that more than 10/15 years of a patient health record is digital.

In this scenario, a special attention should be paid to Internet technologies. To date, the Internet has been used principally as a tool for commerce and as a cost-effective communication medium by the European healthcare industry. Increasingly however, the Internet's potential is being harnessed to transform healthcare delivery at the patient level and is expected to trigger fundamental change in many areas related to healthcare provision, towards what is referred to as e-Health. E-Health is definitely a major goal for any developed country but many barriers are still to be overcome until a worldwide ubiquitous medical information network is achieved. The current situation depends largely on the country but on average in the Member States of the European Union, only 12% of clinicians use information technologies for e-mail consultation and only 2% of patients are enable to book appointments online. Hospitals and other healthcare units usually don't share information between them both for technical and legal reasons, even if clinicians already use email for non-institutional information exchange.

In addition, even where digital records are used and data could be easily analyzed by doctors and prediction tools, the healthcare provision was an inherently distributed process involving many distinct institutions using different equipment, practicing different protocols and cultures. It means that the same analysis might have different digital representation accordingly to the institution where is registered, information might be unreadable between institutions and dangerous misunderstandings might appear if no common or compatible formats are adopted. In other words, standards for data representation, medical protocols and best practices are not yet widely divulged creating heterogeneity on medical data both from the syntactically and semantically.

It is therefore evident that the creation and diffusion of a compatible EHR as well as the adoption of common standards for data representation, storing and transmission is an urgent goal to be pursued.

The adoption and diffusion of EHR will have a significant impact also on the way citizens access their own data. Since in many countries patients have the right to access their personal health record, the availability of electronic health records to patients seems inevitable. On a recent survey 73.8% of patients responded that they know that they have the right to see their health records but only very few (4.5%) have done so. Although the 79% considered a good idea to make health records electronic and the 66% believe that the advantages of electronic records outweigh the disadvantages (23.1% didn't have any opinion) approximately one half were afraid about security (50.4%). It is also important to mention that only 56.4% believed that it would be easy to understand their health records. Patients identified several benefits from accessing their own health records: better understanding of their health, easier plan of consultations, and easier talk to their doctors and better decisions about their own health. The main concerns were security, cost, the need for keeping technology up to date, the distraction of addressing EHR from more important health issues, inability for using computers and the afraid of the content being frightening.

The development of adequate platforms for EHR access of patients can introduce a new dimension on the health record analysis by the patients creating a more citizen centred approach to healthcare. The support given to the patient can be greatly improved with adequate links for explanation of the scientific names and contents, with different views of the information depending on the profile of the user and other support techniques that will improve the usefulness and the diffusion of the health record visualization by the patients. Another important factor is the availability of decision support tools that are designed for physicians but are made available also to patients. Such decision aids can influence decisively the attitudes of the patients reinforcing the recommendations or stimulating the conflict with the doctors leading to self medication attitudes that can negatively influence the outcome of serious diseases such as cancer.

A main, fundamental role to the evolution of the medical sector towards digital based documentation that is developing fast on all other activity sectors, is being plaid and will be plaid in the near future by the healthcare structure, both public and private. These are affording a period of strong renovation and adaptation to the new ICTs with the goal of guaranteeing a more efficient and effective healthcare service and reducing at the same time general costs. It is more and more necessary the capillary introduction of ICTs in order to allow integration and interoperability among healthcare and territorial structures that so far treat the patients' data with logical and physically different approaches.

Nonetheless, according to our experience the healthcare scenario is actually characterised by the following main problems:

- a centralised patient demographic index does not exist; it is often the case that every structure has its own demographic databases, which are redundant and not synchronised, thus making impossible the extraction of the general data of a patient every time he has to be identified.
- a central booking service does not exist, so that the schedules of each single department (e.g. Radiology, Surgery, Echo, etc.) are not synchronised to each other; moreover, even when a department has an ICT booking system, this is not communicating with any central booking system, therefore it is impossible to know which services are actually supplied, if they have been changed and, overall, it is impossible to make a daily booking plan both for the business units and for the patients that usually have to return several times to the healthcare centre.
- There is not automated mechanism to require a service provision, either internal or external, to a service provider; similarly there is no automatic service for drugs ordering.
- There is not automated reporting from the service provider to the service applicants. The use of an ICT reporting system that includes digital signature mechanisms, would allow the applicants to receive the report of an exam at the same time when it was issued and to visualise it immediately anywhere and anytime it is needed.
- It is practically impossible to perform aggregation and comparison between the data available on a patient.
- It is nearly impossible to quantify the consumption on consumable goods, such as drugs, tapes, etc., internal services and, in general, to monitor specific benchmarking indices, that allow the definition of short, mid and long terms strategies of the healthcare structures.

This scenario shows how fragmented and incomplete the Health Information System (HIS) is, where any software system is operating independently, is not inserted in a controlled workflow and is, on the contrary, specifically dedicated to a local activity of the department in which it is being used.
In order to overcome these limits and provide the promised benefits, the HIS should provide the following features:

- The systems should be integrated so that data can be acquired only once and it can be accessed anywhere and anytime, thus reducing costs related to recording, management and control of the available information and the need of repeating some medical tests;
- The systems must be reliable and secure:
    - They should manage user credentials, such as reading, writing, modification rights;
    - Privacy and secrecy should be guaranteed by using adequate encryption mechanisms;
    - Suitable certification mechanisms should be implemented in order to guarantee the originality of the information;
- The systems should reduce as much as possible the production, circulation and archiving of information on traditional supports such as paper, tape, etc., thus moving towards a paperless reality.

The first point, which implies the use of Electronic Records (ERs), is particularly important, although the other aspects cannot be disregarded. In fact ERs allow reducing significantly the management costs, both in terms of human and technical resources. ER also eliminates data redundancy and guarantees the access to up-to-dated information. Moreover, a web-based architecture with distributed data allows the integration of data provided by heterogeneous sources.

According to the Medical Records Institute, five levels of an Electronic HealthCare Record can be distinguished in the follow stages:

- Stage 1 (AMR) - Automated medical record system: a paper-based record with some computer-generated documents.
- Stage 2 (CPR) - Computerised medical record system: makes the documents of level 1 electronically available.
- Stage 3 (EMR) - Enterprise electronic medical record system: restructures and optimizes the documents of the previous levels ensuring interoperability of all documentation systems.
- Stage 4 (EPR) - Electronic patient record system: patient-centred record able to integrate information from multiple EMR.
- Stage 5 (EHR) - Electronic health record: adds general health-related information to the EPR that is not necessarily related to a disease. The main goal of an EHR is to reconstruct the entire clinical history of a patient, starting from his first contact with a healthcare structure up to today and including all the information acquired at each contact with any health structure during his entire life. This is a new vision of the concept related to the digitalisation of patient information: it implies the integration of

information provided by different institutes, located everywhere on the territory, even in different countries.

The first step towards the digitalisation of the clinical data was performed by the introduction of an EMR that is a digital archive containing exactly the same data of paper-based archives.

On the contrary, the EHR is able to integrate all the information collected in the different EPR at each contact of the patient with any healthcare structure operating on the territory, both in the same nation and abroad. This result can be achieved using a single Master Patient Record that is a global identifier able to univocally identify each patient along his entire life (life-long Patient Identifier). The EHR is able to collect all the data acquired by the General Practitioners (GPs), specialists and other healthcare centres, as well as data related to bioinformatics information (genetics, genomics, etc.) or general wellness records.

The use of EHR for the management of clinical data will significantly support the achievement of the following results:

- Continuity of care;
- Integrated "patient-centric" and "disease-centric" vision;
- More effective follow-up of the patients;
- Definition of more adequate health protocols.

One of the main objectives of the HEARTFAID project, will be study and implementation of a specific EPR to be adopted in the cardiovascular context, and in particular in the HEARTFAID validation scenarios, for the acquisition of selected cardiovascular data; later this dedicated EPR will be extended towards what will become a preliminary mock-up of the EHR that will be necessary for the traceability, collection and integration of the data identified by the clinical partners as relevant for the project purposes

5.8 Roadmap: Next Generation Smart Healthcare

It is evident that the next generation smart healthcare will be mainly based on the homecare idea, where the sensor networks play a major role. The future wireless sensor network architecture for smart homecare will possess the essential elements of each of the future medical applications, namely:

- ***Integration*** with existing medical practices and technology,
- Real-time, long-term, ***remote monitoring***,
- ***Miniature***, wearable sensors, and
- ***Assistance to the elderly and chronic patients***.

It extends healthcare from the traditional clinic or hospital setting to the patient's home, enabling telecare without the prohibitive costs of retrofitting existing dwellings. Currently, patients visit doctors at regular intervals, self-reporting experienced symptoms, problems, and conditions. Doctors conduct various tests to arrive at a diagnosis and then must monitor patient progress throughout treatment. In smart homecare, the WSN collects data according to a physician's specifications, removing some of the cognitive burden from the patient (who may suffer age-related memory decline) and providing a continuous record to assist diagnosis. In-home tasks are also made easier, for example, remote device control, medicine reminders, object location, and emergency communication. The architecture is multi-tiered, with both lightweight mobile components and more powerful stationary devices. Sensors are heterogeneous, and all integrate into the home network. Multiple patients and their resident family members are differentiated for sensing tasks and access privileges.

Smart homecare benefits both the healthcare providers and their patients. For the providers, an automatic monitoring system is valuable for many reasons. Firstly, it frees human labor from 24/7 physical monitoring, reducing labor cost and increasing efficiency. Secondly, wearable sensor devices can sense even small changes in vital signals that humans might overlook, for example, heart rate and blood oxygen levels. Quickly notifying doctors of these changes may save human lives. Thirdly, the data collected from the wireless sensor network can be stored and integrated into a comprehensive health record of each patient, which helps physicians make more informed diagnoses. Eventually, the analyzing, diagnosis, treatment process may also be semi-automated, so a human physician can be assisted by an “electronic physician."

Healthcare patients benefit from improved health as a result of faster diagnosis and treatment of diseases. Other quality-of-life issues, such as privacy, dignity, and convenience, are supported and enhanced by the ability to provide services in the patient's own home. Family members and the smart homecare network itself become part of the healthcare team. Finally, memory aids and other patient-assistance services can restore some lost independence, while preserving safety.

Examples of envisioned missions where the WSNs can quickly make an impact are the following:

- *Sleep apnea*. Every night, monitor blood oxygenation, breathing, heart rate, EEG, and EOG using on-body sensors to assess severity and pattern of obstructive sleep apnea. Home network monitors agitation (movement) and stores and reports sensor data. Network alerts provider and patient if oxygenation falls below a threshold. Monitoring can continue while treatment efficacy is assessed.

- *Journaling support*. Journaling is a technique recommended for patients to help their physicians diagnose ailments like rheumatic diseases. Patients record changes in body functions (range of motion, pain, fatigue, sleep, headache, irritability, etc), and attempt to correlate them with environmental, behavioral, or pharmaceutical changes. The homecare network can aid patients by: providing a time-synchronized channel for recording and transmitting the journal (PC, PDA, “dizziness" button); recording environmental data or external stimuli (temperature, barometric pressure, sunlight exposure, medication schedule); and quantitatively measuring changes in symptoms (pain, heart-rate, sleep disruption).

- *Cardiac health*. Cardiac arrhythmia is any change from the normal beating of the heart. Abnormal heart rhythms can cause the heart to be less efficient, and can cause symptoms such as dizziness, fainting, or fatigue. Since they are sometimes very brief, it can be difficult to properly characterize them. Cardiac stress tests attempt to induce the event while the patient is wearing sensors in a laboratory. In a homecare setting, wearable EKG sensors can monitor for the condition continuously, over days or weeks, until the event occurs. The recorded data is promptly sent to the physician for analysis. If the event is serious enough, the emergency communication channel may be used to call for help, or it may be dispatched automatically. Other sensors in the home may be able to record environmental data to help identify the cause (side-effect of medicine, little sleep, etc.).

6 DEFINITION OF THE SYSTEM REQUIREMENTS

6.1 General Requirements

The general requirement of the data acquisition and transmission infrastructure can be simply stated by the phrase: "***Efficient and reliable data collection in all the related healthcare environments using the available sensor equipment and transmission of the data to the HEARTFAID platform using the existing networking fixed and mobile technology***". The first step towards moving more in depth towards decomposing the general requirements into more specific ones is the definition of the healthcare environments, in which the data acquisition and transmission takes place. In the following paragraphs we give a detailed description of these environments, the involved actors and the requirements imposed by each environment.

6.2 Definition of the healthcare environments

The healthcare identified healthcare environments are the following:

i) The clinical environment
ii) The patient's environment, which can be divided into
   a. The in home environment
   b. The on-the-move environment

6.2.1 The Clinical environment

6.2.1.1 Description

With the term "clinical environment" we refer to the scenario in which the patient's biomedical data is acquired in the hospital premises. In a typical scenario, a PC located at the hospital premises plays the role of the device which acquires the data from the sensors and transmits this data to the HEARTFAID platform.

6.2.1.2 Involved "actors"

We can define the following main actors in the hospital scenario:

- The patient
- The sensors / medical devices used in the hospital environment
- The personnel of the hospital, which may operate some of the medical devices
- The device which collects, and transmits the medical data (typically a PC)
- Possible existence of personnel which operates some device related software on the PC

6.2.1.3 Requirements related to the data acquisition and transmission proccess

The main requirements related to the data acquisition and transmission process in this environment are:

i) Use of the sensor equipment available in the hospital premises
ii) Cheap communication from the stock
iii) Sensors/medical devices support for connectivity
iv) Sufficient communication range (in case of wireless sensors)
v) Naming and addressing of devices and intermediate acquisition and transmission elements
vi) Easy incorporation of the devices in the infrastructure (plug and play features)
vii) Software applications for integration of raw data and/or data generated by vendor specific software into the HEARTFAID platform
viii) Secure data transmission
ix) Privacy
x) Quality
xi) Modular acquisition and transmission software design

6.2.2 The in Home environment

6.2.2.1 Description

With the term "In-home environment" we refer to the scenario in which the patient is located in his home and used medical sensors/devices in the home environment. The biomedical data produced by the measurement of the patient's vital signs are collected by a device (typically a PC) located at the patient's home and are transferred to the HEARTFAID platform using and Internet (e.g ADSL) connection.

6.2.2.2 Involved "actors"

In the in-home environment, the main actors are:

- The patient
- The sensors / medical devices used in the hospital environment
- The device which collects, and transmits the medical data (typically a PC)

6.2.2.3 Requirements related to the data acquisition and transmission proccess

The main requirements related to the data acquisition and transmission infrastructure in the in-Home scenario are the following:

i) Use of the sensor equipment which can be easily installed at the patient's home
ii) Use of sensor equipment which affect's the patient's everyday life in minimum degree.
iii) Sensors/medical devices support for connectivity
iv) Cheap communication from the stock

v) Sufficient communication range inside the home in order to allow the patient to carry the sensors around the home (in case of wireless sensors)
vi) Naming and addressing of devices and intermediate acquisition and transmission elements
vii) Easy incorporation of the devices in the infrastructure (plug and play features)
viii) Easy to use software applications (with many automated features) for integration of raw data and/or data generated by vendor specific software into the HEARTFAID platform
ix) Secure data transmission
x) Privacy
xi) Quality
xii) Modular acquisition and transmission software design

6.2.3 The on-the-move environment

6.2.3.1 Description

With the term “on-the-move environment”, we refer to a scenario in which the biomedical data are collected in an outdoor environment from portable medical sensors/devices which the patient carries. The biomedical data are collected by an also portable device and transmitted wirelessly to the HEARTFAID platform over Internet.

6.2.3.2 Which are the “actors”

In the on-the-move environment, the main actors are:

- The patient
- The sensors / medical devices used in the outdoors environment
- The device which collects, and transmits the medical data (typically a PDA, smart phone etc).

6.2.3.3 Requirements related to the data acquisition and transmission proccess

The main requirements related to the data acquisition and transmission infrastructure in the in-Home scenario are the following:

i) Use of the portable sensor equipment
ii) Use of sensor equipment which affect’s the patient’s outdoor activity in minimum degree.
iii) Sensors/medical devices support for wireless (preferably) connectivity
iv) Cheap communication from the stock
v) Sufficient communication in the outdoor environment (in case of wireless sensors)
vi) Naming and addressing of devices and intermediate acquisition and transmission elements
vii) Easy incorporation of the devices in the infrastructure (plug and play features)

viii) Easy to use software applications (with many automated features) for integration of raw data and/or data generated by vendor specific software into the HEARTFAID platform
ix) Secure data transmission
x) Privacy
xi) Quality
xii) Modular acquisition and transmission software design

6.3 Identification of the needed parts and subsystems

To fulfill the aforementioned objectives the following elements are needed

Element	Role
Medical Devices / sensors	Produce the biomedical data to be entered into the HEARTFAID platform
Communication units for the medical devices	Enable the medical devices to transmit their data to a data collection device set up in the relevant healthcare environment
Communication units for the data collection devices	Enable the devices that collect the data to communicate with the medical devices
Software modules executing in the data collection devices	• Enable the collection of the transmitted biomedical data via the implementation of the communication protocols (most of the times vendor specific), or via the parsing of output files generated by vendor specific software. • Enable the enhancement of the collected data with additional information (e.g. time-stamping, user information etc) • Enable encryption of data in order to be transmitted over public networks • Enable the transmission of data
HEARTFAID Electronic Patient Record	Stores the biomedical data and makes it available to other entities

In the following sections, a detailed discussion related to the requirements imposed by the use of each element is given.

6.4 Requirements related to medical devices/sensors

6.4.1 Wired Medical Devices

A fundamental requirement in the modern medical/healthcare industry is to add network connectivity to previously isolated medical devices, in order to provide centralized control and monitoring. Furthermore, the integration of medical devices with information systems automates the workflow surrounding the medical device, and lowers the error rates due to manual data entry and duplication. This workflow may carry out meticulous data analysis or acquire and communicate data generated by devices to other information systems or users.

Many parameters affect the configuration of a medical device network and one should gain a firm understanding of such parameters, before attempting to implement a solution. This section presents an overview of the issues involved in setting up a network of wired medical devices. All aspects are equally important, therefore an in-depth estimation of the inherent trade-offs will be performed, before discussing any definitive approach.

- **Bandwidth**

Different types of communication and information exchange have different requirements in terms of bandwidth. The following table provides a rough estimation of these bandwidth requirements. These figures are an approximation, based on acceptable delays in communication. If for instance, one does not mind waiting a minute for a black and white picture, instead of a couple of seconds, the required bandwidth decreases. Nevertheless, the table provides an indication of the different requirements, and helps in selecting appropriate technology.

Medium	Bandwidth Requirements (approx.)
Text-based communication	0,5 Kb/s
Black & white pictures	20 Kb/s
Colour pictures	50 Kb/s
Voice communication	50 Kb/s
CD quality audio	144 Kb/s
VCR quality video	2 Mb/s
Medium quality video	5 Mb/s
High quality video	20 Mb/s

Table 1 Estimated bandwidth requirements

The immediate interest of HEARTFAID is focused on low bit rate text-based communication, since this medium will be used for the system's implementation. However, HEARTFAID architecture has been drawn with no presumptions on implementation details such as data types or network capacity, and is therefore able to carry and process any type of data, as long as the respective sensors and adapters are configured.

- **Connectors and Transmission Media**

Besides the bandwidth demands of the captured data, the available device connectors are another important factor of connectivity. Today, the most widely used connectivity medium is the RS-232 serial port, which depending on its configuration features may support a bit rate from 20Kbps to 230Kbps. Even though the RS-232-C standard was defined as far back as 1969, it stays very popular today, and it remains the implementation of choice for many medical devices. The serial port standard is a suitable option for the devices used in HEARTFAID.

Other than that, many of the modern medical electronic equipment is concerned with imaging; examples include magnetic resonance imaging scanners and ultrasound machines. Producing and manipulating images, especially in real time, involves handling large amounts of data rapidly, so dedicated connectivity connectors and wiring were developed, to tackle this daunting task. It is not uncommon, therefore, to find equipment in which the backplane and input/output (I/O) connector speeds in the image processing sections are 5 GB/s or even higher. Connectors specifically designed to offer good performance at these speeds are essential and shielding must be of impeccable quality. Backplane connectors need to be able to handle a multitude of high-speed signals, which makes high connection density with minimal crosstalk important. This has been addressed in the latest connectors by designing integral shielding that provides earth continuity. This eliminates the need for ground pins and allows every pole of the connector to be used for signals.

- **Interoperability and Standards**

At the level of integration of devices, the question of interoperability is twofold: the hardware and software level of interconnection. Over 100 vendors of medical devices or systems offer products that enable connectivity. As discussed in the previous section, in the hardware level it is provided via a serial RS-232 port on the device and, in some more sophisticated cases, via an Ethernet interface on the system. At the hardware level, these interfaces have some variability (connector and actual signal lines used, etc.). However, for the most part, they provide a relatively standard method of ‘connecting’.

This hardware standardisation is not replicated from the software perspective. Over 500 different protocols are utilised by these 100 vendors to provide interfaces to their system. In many cases, the protocols transfer primarily clinical data. However, they are also seeing increasing amounts of service data transferred that can be utilised to improve preventative maintenance dramatically. In selected cases, the data may also specify equipment usage as well as supplies usage for the equipment. In addition to the data being variable, each software protocol is typically vendor-specific and, in many cases, a single vendor will have multiple protocols specific to individual devices within their overall product line. The net result of these issues is device interfaces that do exist and do enable connectivity. However, their usage in modern clinical environments is limited by interface cost as well as the wide variation in protocols. Multiple standards organizations have addressed this confusion in interface standards but their success has been limited. In the early 1980s, a group sponsored by the Institute of Electrical and Electronic Engineers, Inc. (IEEE) developed the Medical Interface Bus. The standard provided for both the transport and lower-level protocol aspects, as well as higher-level protocols for transferring information. Even though it was ratified, at the time, medical vendors had their own proprietary solutions so adoption was extremely limited. Another group – the Point-of- Care Connectivity Industry Consortium (POCCIC) – has widespread vendor support and recently completed an updated device interface standard. The acceptance of this standard remains to be seen. The most critical issue impacting its acceptance is the need for vendors to move systems from previously established protocols to the updated protocol standard. However, even if they do, a number of years will be required for the standard to achieve widespread usage.

- **Security and Data Protection**

Preserving the privacy and integrity of medical data is a very important consideration for this project. Although such issues are mainly within the scope of wireless connectivity solutions, wired implementations should also guarantee that patients' data cannot be accessed by any unauthorized source, at any point of the data lifecycle. For this purpose, HEARTFAID has set a very high priority in employing secure data transport and storage schemes. Once device data are routed beyond the immediate host connection (for example, via a LAN to a remote application) security will be maintained to ensure privacy and source authentication. This is especially true given the requirements mandated by the Health Insurance Portability and Accountability Act of 1996 (HIPAA). A robust data encryption algorithm will be used (3DES, RSA, or the like) to certify end-to-end confidentiality in the transmission of the sensitive medical data.

- **Cost Savings**

The tangible cost savings for providing network connectivity to medical devices can be expressed in terms of time savings due to increased efficiency at the patient's bedside. By not having to read record and transcribe information into an EHR system, a considerable amount of time can be saved. A conservative estimate ranges between one and two minutes for each device connected to the respective gateway PC. When considered across an entire facility, like a Hospital, the opportunity cost savings can quickly exceed the cost of the initial investment.

Furthermore, many intangible benefits come in the form of increased completeness of data readings, a reduction in transcription errors, greater availability of information and the ability to increase sampling frequency. While it may be difficult to place an exact monetary value on each of these intangibles, little argument can be made that they directly affect the quality of patient care in a positive way.

6.4.2 Wireless Medical Devices: Satisfying Radio Requirements

Understanding radio protocols and requirements is the first step in successfully using wireless communication in medical applications. A wide variety of medical applications, such as implantable devices and telemetry equipment, are increasingly using wireless communication. And at the heart of this trend is the need for low-power radio application-specific integrated circuits (ASICs). In order to select the appropriate radio ASIC, it is critical to first understand the protocols and requirements needed for the particular medical device. This section provides an overview of the protocols commonly used in medical radio applications. It also reviews the essential factors to consider when selecting a frequency band, topology, and transmission protocol.

- **Common Medical Radio Requirements**

Medical radios are usually driven by two primary requirements. First, medical radios must consume very little power so that they can last on battery power for months or years. Second, medical radios must often be added to other low-cost, small-sized components such as sensors. This consideration requires that these radios be low cost, have a simple design, and need few external components.

Medical applications are usually indoors, which typically translates into a relatively short range over which data must be transmitted. In addition, the radios need to transmit only a small amount of data, and those data are transmitted only infrequently.

A short data range means that the radio design can be kept simple and that the transmitter power can be kept low. The infrequent data transfer means that medical radios can sleep a large portion of the time. Both the short data range and infrequent data transfer of these radios help to minimize power requirements.

Of course, the requirements of a medical wireless system go beyond low power and low cost. Table 2 lists additional requirements that should be considered when reviewing possible wireless system architectures.

REQUIREMENT	QUESTIONS	COMMENT
Environment	Is the indoor environment likely to be crowded with obstacles? Do other transmitters exist?	A noisy environment requires robust data error detection.
Data rate	How much data must be transmitted? How often? How fast?	Higher data rates consume more power and require more-sophisticated designs than low data rates do.
Data range	How far apart will the transmitter and receiver be?	Longer ranges require more transmit power and sensitive receivers than short ranges do.
Power consumption	What type of batteries will be used? How many?	Power consumption is dependent on many factors including transmitter power, carrier frequency, and data rate
Topology	Will data go point-to-point, or point-to-multipoint?	Point-to-point systems are easier to develop and cheaper than point-to-multipoint systems are.
Speed and latency	How time sensitive are the data?	Overhead associated with error detection and correction increases latency.
Interoperability	Does the radio interface with other radios?	If yes, the radios must operate on the same protocol.
Host intelligence	How much data-processing capability is required?	Overhead associated with error detection and correction increases processing power.
Size and system cost	How much room is available for antennas and external components?	Portable applications have little room for external components.
Security	How secure does the transmitted data need to be?	Encryption codes can be added by the host intelligence if necessary.

Table 2 Wireless System Requirements

- **Selecting a Frequency Band**

Several licensed frequency bands exist specifically for medical applications. In addition, many low-cost applications operate in the unlicensed frequency bands known as the industrial, scientific, and medical (ISM) bands. No license is necessary to operate a device in this band.

Most radio ICs are transceivers, which means the IC can both transmit and receive data. A transceiver is often paired with a simple microprocessor or other support circuitry that directs the baseband operations, such as controlling the flow of information between the real-world interface and the transceiver.

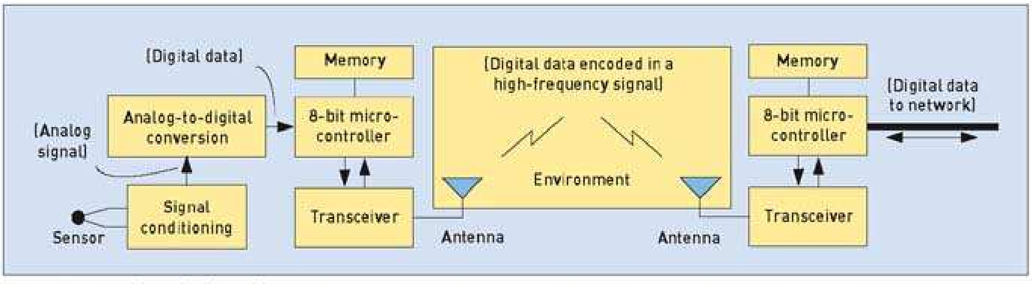


Figure 2 A simple medical wireless system

In the simple wireless system shown in Figure 2, the analog data from a sensor measurement is first converted from its continuous-time analog signal into a discrete-time digital signal. A baseband processor then adds some error-detection information and formats the data for transmission. The baseband data are sent to the transceiver, encoded onto a radio-frequency (RF) signal and broadcast by the transmitter, detected by the receiver, and finally decoded back into digital data. Low-power transceiver ICs usually operate between 300 MHz and 1 GHz. The frequency spectrum between these limits contains many bands where both licensed and unlicensed medical and industrial equipment can operate (see Figure 3). Table 3details the common North American and European frequency bands that are suitable for certain medical applications.

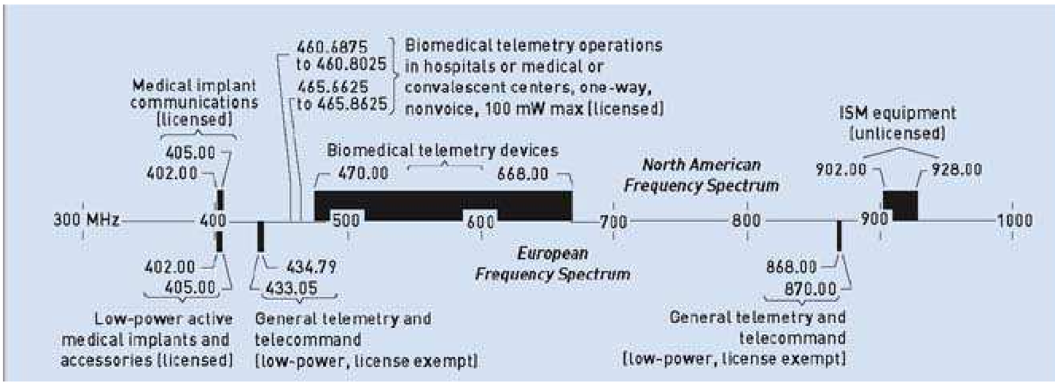


Figure 3 The 300 MHz to 1 GHz frequency spectrum

In the United States, the Federal Communications Commission (FCC) has set aside a few frequency bands for medical data transmissions only. In 1999, FCC set aside the medical implant communications service (MICS) frequency band. The MICS band, located in the frequency range of 402–405 MHz, is reserved specifically for wireless data communications between implanted medical devices and external equipment.

Wireless medical telemetry enables monitoring equipment to remotely and unobtrusively observe several patients at one time. Such telemetry systems transmit real-time physiologic data, so it is critical to ensure that data are not lost or delayed. More and more radios for non-medical applications are operating in the ISM bands, increasing the likelihood of signal loss and interference. FCC, therefore, has set aside frequency bands specifically for wireless medical telemetry services (WMTS). The radio spectrum designated for WMTS includes the 6 MHz between 608 and 614 MHz.

FREQUENCY (MHz)	GEOGRAPHIC REGION	BAND	REGULATION
402.00 to 405.00	North America, Europe	MICS band (licensed)	FCC, 7 CFR 95.601-673; EN 301 839
433.05 to 434.79	Europe	General telemetry band (unlicensed)	EN 300 220
868.00 to 870.00	Europe	General telemetry band (unlicensed)	EN 300 220
602.00 to 614.00	North America	WMTS band	—
902.00 to 928.00	North America, Australia, South America	915 MHz ISM band (unlicensed)	FCC 15.247
2400.00 to 2483.50	Europe	2.4 GHz ISM band (unlicensed)	ETS 300 328
2400.00 to 2500.00	North America	2.4 GHz ISM band (unlicensed)	FCC 15.247
5650.00 to 5925.00	North America	5.7 GHz ISM band (unlicensed)	FCC 15.247

Table 3 Medical frequency bands

- **Topology**

Wireless receivers and transmitters are often arranged in a star configuration, in which a centrally located transceiver communicates with one remote location (a point-to-point topology) or with several remote locations (a point-to-multipoint topology) simultaneously. In a point-to-point architecture, only one transmitter and receiver pair is communicating at any given time on a specific carrier frequency. The central node takes the role of a master coordinator, while the remote location is a subordinate. Point-to-point communications can be simplex (one way only), half duplex (first one direction and then the other, sequentially), and full duplex (simultaneous communication in both directions).

Networks that are more complex may have multiple transmitters and receivers that can communicate with one another as peers. Within a network, one of the central nodes is designated as a coordinator. The coordinator is tasked with waking up other subordinate devices on the network out of a low-current sleep mode just before data are to be transmitted. Coordinator transceivers can also talk to one another as peers.

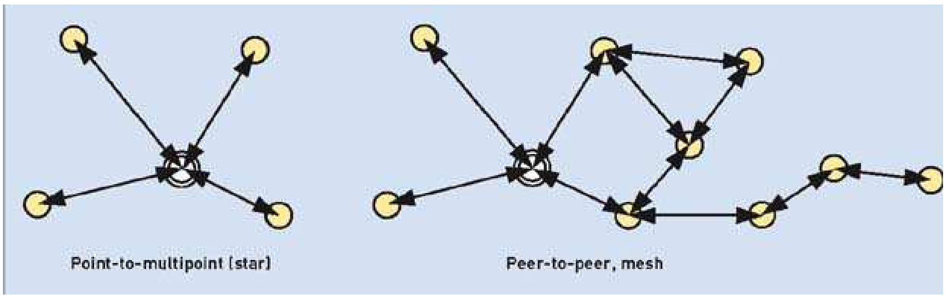


Figure 4 Network topologies

Mesh networks allow wireless devices to talk indirectly to one another even when the two devices cannot see each other. A transmitting device can pass data to its neighbor, which in turn can pass data onto its next neighbor, and so on. In this way, a mesh network can be used over a far-flung network, say, between the top and bottom floor in a high-rise building. Figure 4 illustrates two different network topologies.

- **Transmission Protocols**

Transmission protocols define the way data are to be encoded, transmitted, received, and decoded. Protocols also define any error detection methods as well as techniques to minimize interference and distortion. Protocols are usually optimized for the given application to reduce the overhead associated with error correction and thus result in lower-cost and lower-power operation.

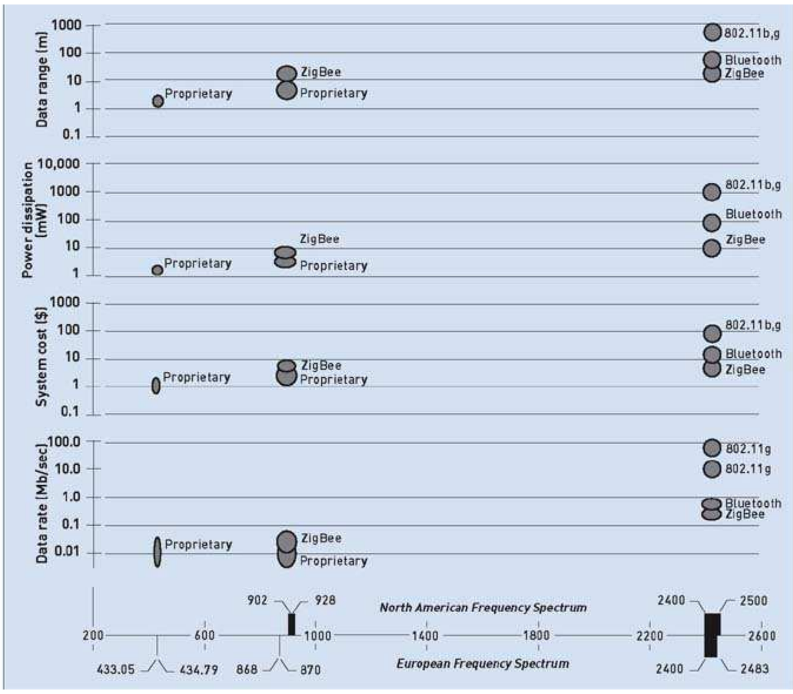


Figure 5 Various protocol features mapped to unlicensed frequency spectrum

Medical equipment manufacturers concerned about the interoperability among radios from different manufacturers should consider using an open-standard protocol such as Bluetooth or ZigBee. Figure 5 maps several open standards against the unlicensed frequency bands. Open-standard protocols can also allow for a complex networking of transmitters and receivers such as those operating in a mesh-type network.

However, standard protocols give up some flexibility, and considerable overhead is required to ensure compatibility and interoperability. Bluetooth and ZigBee radios are also designed for higher-frequency bands, such as the 2.4 GHz ISM band. The wireless local-area-network (WLAN) and wireless personal-area-network (WPAN) open standards were developed to address high-speed data transmission requirements. Because medical applications typically require only low data rates, the WLAN and WPAN standards (e.g., IEEE 802.11a, b, or g) are rarely appropriate for small implantable medical devices.

	TRANSCEIVER A	TRANSCEIVER B
Frequency band	868–870 MHz and 902–928 MHz	868–870 MHz and 902–915 MHz
Protocol	Open standard: IEEE 802.15.4 ZigBee	Proprietary
Topology	Multipoint, peer-to-peer	Point-to-point
Environmental Requirements		
Carrier modulation	BPSK (biphase shift key)	ASK-OOSK (amplitude shift key–on-off shift key)
Multipath reduction technique	Direct sequence spread spectrum (902–928 MHz channels only)	Diversity antennas with dual narrowband receiver chains
Data integrity	16-bit CRC and CSMA/CD	Checksum and ARQ
Data Range		
Transmitter power	0 dBm	+6 dBm
Receiver sensitivity	–92 dBm	Sniff mode: –90 dBm Receive: –103 dBm
Data rate	20 Kb/sec (868–870 MHz) 40 Kb/sec (902–928 MHz)	User programmable from 1.2 to 19.6 Kb/sec
Data range	Up to 100 m	100 to 500 m
Power Consumption		
Current consumption	Receive □25 mA @ 3 V Transmit □14 mA @3 V Sleep: □10 µA	Receive □8 mA Transmit □50 mA Sleep mode with automatic wake-up: 0.75 mA (@ 10% duty cycle) Standby: 500 nA
Supply voltage	3 V ± 0.3 V	2.4 to 3.6 V
Host Intelligence		
Parallel data interface	8-bit multiplexed data/address	None
Serial data interface	RS-232	Two-wire I⁲C-bus
Code complexity	<5 Kbyte	< 1Kbyte
System cost		
Price at volume	$4.75	$2.15
System cost (BOM)	<$6.50	<$4.25
Package	64-PLCC (9 ¥ 9 mm)	18-SOIC (7.62 mm)

Table 4 Open-standard protocol versus proprietary protocol

In some cases, proprietary protocols can fit a given medical application better than a standard. Often, proprietary protocols are the only ones available for MICS band operation. A proprietary protocol can offer more flexibility and provide lower-power operation if interoperability is not a concern. A proprietary protocol can also be less expensive overall.

A few considerations should be taken into account when choosing a protocol and an associated frequency band. Among these considerations are:

- Geographic location. The medical frequency bands used worldwide do not necessarily coincide from country to country (see Table 3). The 2.4 GHz band is generally available in most locations, but it may be restricted depending on the country. Other, lower-frequency bands are restricted to a geographic location.

- Interoperability. If the application requires that devices from several manufacturers must work together, then a specific communications standard must be used. That standard will define the frequency range. The newly adopted IEEE 802.15.4 standard, commonly known as ZigBee, addresses the 868 MHz, 902–928 MHz, and 2.4 GHz ISM bands and ensures interoperability between various radio devices.
- Interference. Multiple wireless standards often coexist in unlicensed frequency bands. For example, 900 MHz cordless phones operate in the 902–928 MHz band. In addition, the 2.4 GHz band is usable in most locations, making it a popular choice for a variety of wireless communications protocols. Significant microwave oven operation also can plague that band.

Low-data-rate radios can usually host both the protocol and application control in a simple 8-bit microcontroller. Typical applications use only between 4 and 30 Kbyte of code and less than 300–4000 bytes of RAM, depending on whether the radio is a full-function controller or a reduced-function device.

By comparison, more-sophisticated radios, such as Bluetooth, require a dedicated core to manage the baseband protocol stack. They also need a separate host device to manage the logical link control and adaptation protocol, protocol interfaces, and applications. In order to achieve this, between 100 and 200 Kbyte of code and about 150 Kbyte of ROM may be required for a full software stack.

A clear understanding of the trade-offs between open standards and proprietary protocols can be developed by comparing two similar radio transceivers that operate in the same 868 MHz (EU) and 902 MHz (U.S.) unlicensed medical bands. Table 4 shows how two different transceiver designs operating in the same frequency bands address the wireless system requirements noted in Table 2. Transceiver A follows a more-complex open-standard protocol (IEEE 802.15.4), while Transceiver B follows a simpler but proprietary protocol.

- **Environmental Requirements**

In a medical environment, data transmissions through a wire or along a circuit board trace are always vulnerable to interfering signals, coupled noise, electromagnetic interference, and noisy or shifting power and ground planes.

A common method of overcoming transmission problems is to use the digital data to modulate a much higher frequency signal, called a carrier signal. Carrier modulation techniques convert the digital signals into frequencies that are inherently more immune to noise.

Because the shifting carrier signal is an ac signal with no ground reference, the carrier signal is not affected by unstable or noisy ground planes. The carrier frequencies can be tuned specifically to avoid electrical noise, and the receiver circuits can be designed to reject out-of-band noise. The receiver can also be designed to reliably detect very small changes in frequency that can yield long data ranges with very little power.

- **Carrier Modulation Techniques**

A number of techniques can encode digital data by varying the amplitude, the frequency, or the phase of a carrier signal. These modulation techniques include:

- Frequency shift key (FSK) modulation, which uses two different carrier frequencies to represent the logic high and logic low of digital data.
- Amplitude shift key (ASK) modulation, which uses one frequency but varies the amplitude of the RF carrier to represent the logic high and logic low of digital data. A variation of ASK is called on-off shift key (OOSK) modulation, where one of the two amplitudes is zero. The OOSK signal is a series of alternating-frequency bursts and quiet periods representing the logic high and logic low of digital data. ASK is more susceptible to noise than FSK, but it is often easier to implement and detect.
- Phase shift key (PSK) modulation, which encodes the digital data by altering the phase of the carrier frequency. Biphase shift key (BPSK) modulation shifts the phase of the RF carrier by 180 degrees in accordance with a digital bit stream.

- **Multipath Propagation Problems**

Once the digital data have been encoded as an RF carrier signal, environmental obstacles can affect the data signal path. Multipath propagation occurs when an RF signal takes different paths when propagating from a transmitter to a receiver. While the signal is en route, obstacles such as walls, chairs, desks, and other items get in the way and cause the signal to bounce in different directions. Therefore, some of the signal encounters a delay, increasing the length of path to the receiver.

Multipath delay causes the information symbols represented in a radio signal to overlap, which confuses the receiver. This is often referred to as intersymbol interference (ISI). Because the shape of the signal conveys the information being transmitted, the receiver will make mistakes when demodulating the signal's information. If the delays are great enough, bit errors occur. The receiver won't be able to distinguish the symbols and interpret the corresponding bits correctly.

- **Radio Interference Problems**

The proliferation of many types of common wireless devices in the unlicensed ISM bands can corrupt the data signals of medical radios operating in the same band. Airborne radiolocation systems also share these bands.

When these unwanted RF signals transmit at the same carrier frequency as a medical radio signal, they can cause an interruption or corruption of that signal. Corrupted signals arrive with errors, missing bits, or not at all, and the receiver cannot recover the data. Depending on the sophistication of the receiver-transmitter set, the transmitter may be asked to resend the data, which in turn adds overhead to the network and causes delays.

- **Multipath and Interference Reduction**

Depending on the application, radio interference may be a minimal concern if the environment is relatively quiet. However, indoor short-range radios must be able to account for multipath problems, especially in the ISM bands.

Both active and passive techniques are used to account for multipath fading. Passive techniques generally cost much less. These techniques rely on diversity antenna systems—two antennas for each radio—which increases the odds of receiving an uncorrupted signal.

Diversity antennas are physically separated from both the radio and from each other to ensure that one of the antennas will encounter fewer multipath propagation effects than the other antenna. The receiver uses two separate receiver paths to filter the signal and then decides which signal is the best choice for demodulation.

Active techniques employ spread-spectrum technology to address multipath fading and RF interference. Two commonly used techniques are direct-sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS). DSSS mixes a data signal with a constantly changing pseudorandom noise signal that spreads a continuous data signal over a wide range of frequencies. This leaves enough room for lower-frequency elements of the DSSS signal to reflect off obstacles much differently than the higher-frequency elements of the signal.

FHSS transmits an intermittent data signal for a short period at one frequency and then hops to other frequencies in a pseudorandom pattern. FHSS avoids interference from other radios by hopping on narrow channels over a wide range of frequencies; transmitting on strong, clear channels; and avoiding noisy or faded ones. Frequency hopping is generally regarded as having better noise immunity than DSSS; however, the data rate is lower.

- **Data Integrity Checks**

Error detection and correction are also important in mitigating the effects of interference and multipath fading. The receiver can check for the presence of errors through an error detection process. Basic designs use a simple checksum, but a cyclic redundancy code (CRC) check is necessary for more-reliable data transmission. With the latter process, the transmitting transceiver adds a CRC to every data packet that is sent. A data packet containing an error will not compute correctly. The drawback of added overhead is more than offset by ensuring the received data are accurate. For low-data-rate radios, a CRC of 16 bits is adequate; for high-data-rate radios, a CRC of 24 bits is ideal.

In a point-to-point topology, a simple handshake protocol between the receiver and transmitter known as automatic retransmit request (ARQ) requires the receiver to send an acknowledgment if data have been received with no errors. If an acknowledgment is not received by the transmitter, the transmitter resends the data.

With complex network topologies, such as point-to-multipoint or peer-to-peer topologies, the protocol must enable one receiver to accommodate multiple transmitters broadcasting at the same time. The primary scheme is called carrier sense multiple access with collision detection (CSMA/CD). CSMA/CD is a set of standard rules that determine how network devices should respond when two devices attempt to use the same channel. The handshaking process between receiver and transmitter is similar to ARQ. After detecting a collision, a transmitter waits a random delay time and then attempts to retransmit the data. If the transmitter again detects a collision, the device waits twice as long to try to retransmit the message. This is known as exponential back-off.

Because of data retransmissions, the data rate throughput is slow when multi-path propagation is significant. The reduction in throughput depends on the environment. Radio signals in homes and offices may encounter 50 nanoseconds of multi-path delay. Metal machinery and racks in a plant, however, provide a lot of reflective surfaces for RF signals to bounce from and take erratic paths. Signal delay in a manufacturing plant could be as high as 300 nanoseconds. It is essential to be aware of multi-path problems in warehouses, processing plants, and other areas full of irregular, metal obstacles.

6.5 Requirements Related to Sensor Communication

6.5.1 Wired Technologies

When producing network aware devices, the main focus of medical device manufacturers is in wired connectivity – usually combined with a proprietary communication protocol. There is a growing trend to switch to wireless connectivity in new devices, but even so a large number of highly specialized medical equipment is still using solid network cables for communication and integration with centralized IT systems.

Putting aside the fact that software standardization is a long way from healthcare applications, hardware standardization has been relatively successful in recent years. The vast majority of wired medical devices use the RS-232 serial port as the physical transmission layer (Layer 1 of the OSI Reference Model), while the versatile USB2.0 standard is also moving out of a market niche and into a mainstream implementation. Both of these standards are examined in an effort to identify the optimal device connectivity solution for HEARTFAID. However, the comparison is not completely unbiased; the ***RS-232 has an advantage, since its use is extremely more widespread***.

6.5.1.1 RS-232 Standard

There has been a short mention of this standard's technical profile in section 6.4.1, in the discussion on hardware interoperability. The RS-232 will be examined much more thoroughly in this section, to highlight its description, requirements, technical specifications and limitations.

- **Bi-Directional Communications**

The serial port is a full-duplex device meaning that it can send and receive data at the same time. In order to be able to do this, it uses separate lines for transmitting and receiving data. Some types of serial devices support only one-way communications and therefore use only two wires in the cable - the transmit line and the signal ground.

RS-232 stands for Recommend Standard number 232 and C is the latest revision of the standard, defined in 1969. The serial ports on most computers use a subset of the RS-232C standard. The full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins.

- **DCE and DTE devices**

Two important terms in relation to the RS-232 standard are DTE and DCE. DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. These terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins. The computer is a DTE device, while most other devices, including medical sensors used in HEARTFAID, are DCE devices.

- **Baud vs Bits per Second**

Baud really refers to the modulation rate or the number of times per second that a line changes its state. This is not always the same as bits per second (BPS). If you connect two serial devices together using direct cables then baud and BPS are in fact the same. Thus, if you are running at 19200 BPS, then the line is also changing states 19200 times per second.

- **Cable Length**

The RS-232C standard imposes a cable length limit of 50 feet. However, this limitation may be safely ignored, since a cable can be as long as 10000 feet at baud rates up to 19200 if a high quality, well shielded cable is used. The external environment has a large effect on lengths for unshielded cables. In electrically noisy environments, even very short cables can pick up stray signals. The following chart offers some reasonable guidelines for 24 gauge wire under typical conditions. One can greatly extend the cable length by using additional devices like optical isolators and signal boosters. Optical isolators use LEDs and Photo Diodes to isolate each line in a serial cable including the signal ground. Any electrical noise affects all lines in the optically isolated cable equally - including the signal ground line. This causes the voltages on the signal lines relative to the signal ground line to reflect the true voltage of the signal and thus cancelling out the effect of any noise signals. A relative comparison of the cable lengths vs. baud rate is presented in Figure 6 RS-232 Cable Length.

Baud Rate	Shielded Cable Length	Unshielded Cable Length
110	5000	1000
300	4000	1000
1200	3000	500
2400	2000	500
4800	500	250
9600	250	100

Figure 6 RS-232 Cable Length

- **Limitations of the Standard**

Since the application of RS-232 has extended far beyond the original purpose of interconnecting a terminal with a modem, successor standards have been developed to address the limitations. Issues with the RS-232 standard include:

- The large voltage swings and requirement for positive and negative supplies increases power consumption of the interface and complicates power supply design. The voltage swing requirement also limits the upper speed of a compatible interface.
- Single-ended signalling referred to a common signal ground limit the noise immunity and transmission distance.
- Multi-drop (meaning a connection between more than two devices) operation of an RS-232 compatible interface is not defined; while multi-drop "work-arounds" have been devised, they have limitations in speed and compatibility.
- Asymmetrical definitions of the two ends of the link make the assignment of the role of a newly developed device problematic; the designer must decide on either a DTE-like or DCE-like interface and which connector pin assignments to use.
- The handshaking and control lines of the interface are intended for the setup and takedown of a dial-up communication circuit; in particular, the use of handshake lines for flow control is not reliably implemented in many devices.
- While the standard recommends a connector and pin out, the connector is large by current standards.

6.5.1.2 USB 2.0 Standard

The Universal Serial Bus (USB) is an extremely flexible, standardized connector, used to link up a great variety of external devices to a PC. USB makes plugging in new peripherals easy with plug and play, it is nearly 100 times faster than the original serial port, and it supports multiple device connectivity. Because of these benefits, USB is enjoying broad market acceptance today.

Hi-Speed USB (officially defined 2.0) extends the speed of the connection from 12 Mbps on Original USB up to 480 Mbps on Hi-Speed USB, providing an attachment point for next-generation peripherals which complement higher performance PCs and user applications. Hi-Speed USB is both forward and backward compatible with Original USB, resulting in a seamless transition process for the end user. In fact, Hi-Speed USB uses the same cables and connectors as Original USB. Hi-Speed USB offers a compelling opportunity for peripherals vendors to migrate their USB peripherals to higher performance, while still being able to sell the same peripherals into the huge installed base of USB-capable PCs.

Supporting three speed modes (1.5, 12 and 480 megabits per second) USB 2.0 supports low-bandwidth medical devices, as well as high-bandwidth ones, depending on the functionalities of these devices. The transmission speed of USB 2.0 also facilitates the development of next-generation PCs and applications. In addition to improving functionality and encouraging innovation, USB 2.0 increases the productivity of user applications and allows the user to run multiple PC applications at once or several high-performance peripherals simultaneously. The endless capabilities of the USB connector have not escaped the attention of the medical device industry, which is slowly embracing this technology.

- **Technical Details**

USB connects several devices to a host controller through a chain of hubs. In USB terminology devices are referred to as *functions*, because each individual physical device may actually host several functions, The hubs are special purpose devices that are not officially considered functions. There always exists one hub known as the root hub, which is attached directly to the host controller.

These devices/functions (and hubs) have associated *pipes* (logical channels). The pipes are synonymous to byte streams. Pipes are connections from the host controller to a logical entity on the device named an *endpoint*, in our case, the medical sensors. The term *endpoint* is also occasionally used to refer to the entire pipe. A device/function can have up to 32 active pipes, 16 into the host controller and 16 out of the controller.

The pipes are also divided into four different categories by way of their *transfer type*:

- *control transfers* - typically used for short, simple commands to the device, and a status response

- *isochronous transfers* - at some guaranteed but with possible data loss, e.g. real-time audio or video
- *interrupt transfers* - devices that need guaranteed quick responses, e.g. pointing devices and keyboards
- *bulk transfers* - large sporadic transfers using all remaining available bandwidth (but with no guarantees on bandwidth or latency), e.g. log file transfers from the medical sensors.

When a device (function) or hub is attached to the host controller through any hub on the bus, it is given a unique 7 bit address on the bus by the host controller. The host controller then polls the bus for traffic, usually in a round-robin fashion, so no device can transfer any data on the bus without explicit request from the host controller. The *interrupt transfers* on corresponding medical sensors do not actually interrupt any traffic on the bus: they are just scheduled to be queried more often and in between any other large transfers, thus "interrupt traffic" on a USB bus is really only high-priority traffic.

- **Host Controllers**

The hardware that contains the host controller and the root hub has an interface geared toward the programmer which is called *Host Controller Device* (HCD) and is defined by the hardware implementer. In practice, these are hardware registers (ports) in the computer.

The USB 2.0 HCD implementation is called the *Extended Host Controller Interface* (EHCI). Only EHCI can support hi-speed transfers. Each EHCI controller contains four virtual HCD implementations to support Full Speed and Low Speed devices.

- **Device Classes**

Devices that attach to the bus can be full-custom devices requiring a full-custom device driver to be used, or may belong to a *device class*. These classes define an expected behaviour in terms of device and interface descriptors, so that the same device driver may be used for any device that claims to be a member of a certain class. An operating system is supposed to implement all device classes so as to provide generic drivers for any USB device. Windows XP, which will be used as the operating system of the home gateway, support the implementation described above.

- **USB Connectors**

The original USB specification defined a “Series A” and “Series B” connector, as shown in Figure 7, below.

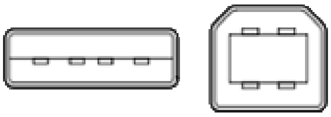


Figure 7 "Series A" and "Series B" Connectors

Cables have only plugs, and hosts and devices have only receptacles. Hosts, like the "InHome" client server have type-A receptacles, whereas the medical devices have type-B, thus making it impossible to err when plugging a device into a USB port. When a USB enabled medical device is connected to the client server PC, the Windows XP operating system will auto-detect it and will prompt for the installation of the drivers of the medical sensor device. If the device has already been installed, the computer activates it and starts interacting with it. USB medical devices can be connected and disconnected at any time.

6.5.2 Wireless Technologies

With advances in wireless networks and portable handheld devices, it becomes natural to use such technologies to improve the effectiveness of health service providers. These technologies enable the collection of biomedical data and the remote monitoring of patients regardless of their location. Over the last several years, wireless technologies have made significant progress, and they are now being integrated into many mainstream applications. In particular, Bluetooth is now seeing increased use in a variety of medical applications ranging from home-healthcare devices to operating room equipment.

Designers have alternative technologies available for wireless connectivity, and their choice is often based on the intended use of the device. In addition, consideration to the new attributes that are inherent in wireless must be considered: **usability, power, distance, data rates, and coexistence**. It is helpful to compare a few major characteristics to understand how to determine when to use which technology. Bluetooth, for example, offers the **lowest power consumption** of all of the networked technologies. It typically requires 1/10 to 1/5 the power of IEEE 802.11b solutions. Typical PDA implementations yield 6–10 hours of usage compared with 2–4 hours for 802.11 solutions using the same batteries. With Bluetooth, no IP addressing is involved, so it is relatively quick and easy to set up small networks of devices. Unlike IrDA, Bluetooth can be used in a small network.

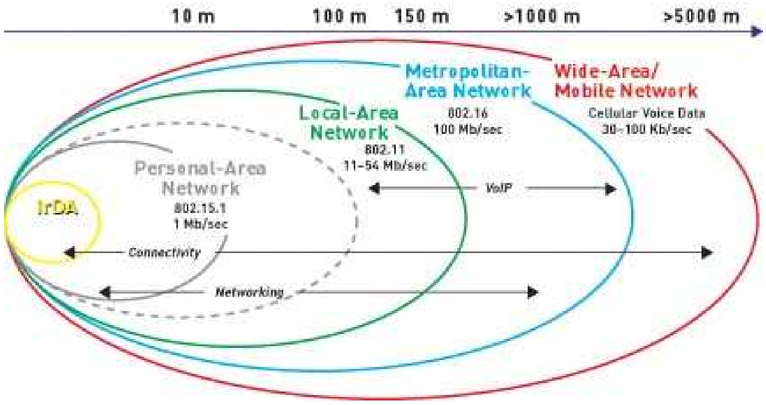


Figure 8 . Industry standard wireless technologies: applications and range

When appropriately used, Bluetooth provides security that meets the needs of HIPAA for patient data. Security requirements in the approved IEEE 802.11 standards raise issues that have yet to be addressed in a standardized way. Unfortunately, many designs produce incompatibilities when connections are implemented between vendors. Bluetooth has yielded to attacks only in products that have not implemented security appropriately or that leave security turned off by default. With the exception of IrDA, **Bluetooth is the most cost-effective wireless technology** currently available. For this reason, the major characteristics and specifications of Bluetooth Technology are illustrated in more detail in the following paragraphs.

6.5.2.1 Key Specifications and Features of Bluetooth Technology

Bluetooth is an industrial specification for wireless personal area networks (PANs). It was first developed in 1994 by Sven Mattisson and Jaap Haartsen, who were working for Ericsson Mobile Platforms in Lund, Sweden at the time. The specifications were formalized by the Bluetooth Special Interest Group (SIG). The SIG was formally announced on May 20, 1998. Today it has over 6000 companies worldwide. It was established by Ericsson, Sony Ericsson, IBM, Intel, Toshiba and Nokia, and later joined by many other companies as Associate or Adopter members. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, PCs, printers, digital cameras and video game consoles via a secure, globally unlicensed short-range radio frequency.

Bluetooth Versions 1.0 and 1.0B had many problems and the various manufacturers had great difficulties in making their products interoperable. 1.0 and 1.0B also had mandatory Bluetooth Hardware Device Address (BD_ADDR) transmission in the handshaking process, rendering anonymity impossible at a protocol level, which was a major setback for services planned to be used in Bluetooth environments, such as Consumerium. Bluetooth Version 1.1 fixed many errors found in Version 1.0B, while it added support for non-encrypted channels and also included a received signal indicator (RSSI). Bluetooth Version 1.2 is backwards compatible with 1.1 and the major enhancements include

Adaptive Frequency-hopping spread spectrum (AFH), higher transmission speeds in practice, extended Synchronous Connections (eSCO), host Controller Interface (HCI) support for 3-wire UART and HCI access to timing information for Bluetooth applications. After Version 1.2, Version 2.0 was released. This version is backwards compatible with 1.x. The main enhancement is the introduction of Enhanced Data Rate (EDR) of 3.0 Mbps, allowing 3 times faster transmission speed (up to 2.1 Mbit/s), 100 meter range, lower power, simplification of multi-link scenarios due to more available bandwidth, further improved BER performance. Finally, Version 2.1 is the latest version of Bluetooth (a draft version of the Bluetooth Core Specification Version 2.1 + EDR is now available from the Bluetooth website).

- **Bluetooth Uses**

Bluetooth is a radio standard and communications protocol primarily designed for low power consumption, with a short range (power class dependent: 1 meter, 10 meters, 100 metres) based around low-cost transceiver microchips in each device. Bluetooth lets these devices communicate with each other when they are in range. The devices use a radio communications system, so they do not have to be in line of sight of each other, and can even be in other rooms, so long as the received transmission is powerful enough.

Class	Maximum Permitted Power (mW)	Maximum Permitted Power (dBm)	Range (approximate)
Class 1	100 mW	20 dBm	~100 meters
Class 2	2.5 mW	4 dBm	~10 meters
Class 3	1 mW	0 dBm	~1 meter

- **Transmission Channels**

Two types of transmission channels are defined in Bluetooth: asynchronous communications link (ACL) and synchronous connection oriented (SCO) link. Since SCO link is defined for the transmission of voice data, its usage is not in the scope of HEARTFAID, for this reason only the details regarding ACL link are given below.

The ACL link is used for data communications and is set up for every link between two Bluetooth devices. It is a packet-switched transmission method that provides error detection, forward-error correction, packet tracking (numbering), and packet retransmission. It is an error-free link to the application; it will either give good data or keep trying to get good data until it signals that the link is down. It is important to note that it is the ***retransmission of data in the presence of interference that increases latency and slows down net data rates***.

The SCO and ACL channels share the total bandwidth, and so a combination of the two types of channels can be used in a piconet. The total bandwidth can include multiple SCO and ACL channels. The availability of two types of channels allows users to select the packet length and to determine the amount of forward-error correction. These parameters are often automatically controlled depending on the desired data throughput along with the amount of interference the device encounters.

- **Piconets and Scatternets**

A Bluetooth piconet consists of at least one master and one slave; this is defined as a point-to-point connection. A full piconet consists of one master and ***up to seven slaves*** (eight total devices). The master controls all timing, including the clock and hopping sequence (to which all slaves synchronize). Each master has a slightly different clock, or skew, and hopping sequence, which is based on both its device address (48-bit IEEE address, as used for Ethernet) and when the device is powered on. These differences allow for multiple piconets to be established and used in the same physical space. A Bluetooth master is responsible for controlling all data traffic in a piconet. All transmissions go through the master in a ***star network topology***. This topology does affect the realizable bandwidth for any given application and configuration. The PAN profile allows devices to talk to each other without knowing that they are in the configuration; the master seamlessly coordinates the transfer of data.

The Bluetooth specification defines the ability to exchange the master and slave relationship between two devices. It also allows a device to be both a master on one piconet and a slave on another, or to be a slave on more than one piconet if the hardware and base-band implementations support it.

A scatternet is formed from two or more piconets that share a common member. This shared member may either be a slave on both piconets or a master on one and a slave on another. Each of these configurations has architectural trade-offs, and the Bluetooth 1.1 specification does not define the preferred one, nor does it completely define the operation of scatternets.

- **Bluetooth profiles**

Collections of features and functionality required to perform a given function are called profiles for Bluetooth devices. The required profile gives all devices some level of interoperability—the ability to function automatically with other devices with little user intervention. These required functions are called the generic access profile. Additional optional profiles depend upon the application requirements and implementation details. There is a big list of profiles currently adopted by the Bluetooth SIG. It is important to note that a device can support multiple profiles. Additional profiles continue to be released independently of the main specification. For the sake of simplicity, a description is given only for profiles relative to the use of Bluetooth in the scope of HEARTFAID):

  • **Device ID Profile (DID):** This profile allows a device to be identified above and beyond the Device Class according to the Specification version met, the Manufacturer, product and product version. This could be useful in allowing a PC to identify a connecting device, and download appropriate drivers. It enables similar applications to those the Plug-and-play specification allows.

- **Personal Area Networking Profile (PAN):** This profile is intended to allow the use of Bluetooth Network Encapsulation Protocol on Layer 3 protocols for transport over a Bluetooth link.
- **Serial Port Profile (SPP):** This profile is based on the ETSI TS07.10 specification and uses the RFCOMM protocol. It emulates a serial cable to provide a simply implemented wireless replacement for existing RS-232 based serial communications applications, including familiar control signals. It provides the basis for DUN, FAX, HSP and AVRCP profiles.
- **Service Discovery Application Profile (SDAP)**: This mandatory profile is used to find out which profiles are offered by the Server device.

6.6 Requirements related to data collection devices

In the Clinical and In-home scenario, the role of the data collection device could be played by a PC, which will be responsible to communicate with the medical sensors / devices in its area (using wireless or wired connections – as described earlier). This PC is responsible to acquire the data, enhance them and transmit them to the platform. Exactly the same role will be played in the on-the-move scenario by a mobile device (e.g. a smart phone). In the following paragraphs we describe the requirements related to the data collection devices and present in parallel some key aspects and technologies related to the specific topic.

6.6.1 Wired devices (Home and Clinical Scenario)

It has been illustrated in previous sections that modern medical sensors, whether wired or wireless, use fairly standardized connectivity hardware. This gives a considerable advantage to the project, since interoperability and modularity are considered very important. By interfacing the sensor with any common personal computer, HEARTFAID sets almost no limit to the number of people that can benefit from it. In essence, all one needs to participate is a PC, and of course the sensor device. The user interacts with the Front-End, a Java-based application, which may be ported and ran in any platform that supports Java. The user may select any operating system that suits her needs, with the single requirement that a JVM can run on the selected OS. There is a very long list of operating systems for personal computers, but very few stand out as considerable market players. An overview of operating systems follows, as well as a deeper examination of the most popular products.

6.6.1.1 Operating Systems

An operating system (OS) is a computer program that manages the hardware and software resources of a computer. At the foundation of all system software, an operating system performs basic tasks such as controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating networking, and managing files. It also may provide a graphical user interface for higher level functions. It forms a platform for other software. The OS provides the user interface for the interaction between the human and the computer, performs job management, task management as well as data and device management operations.

o *Common Operating Systems*

The primary operating systems in use are the many versions of Windows (from 95 to Vista), Macintosh OS X, the many versions of Linux and Unix, OS/400 (IBM iSeries) and z/OS (IBM zSeries mainframes). DOS is still used for some applications, and there are other special-purpose operating systems. An overview of the most widely used and significant desktop operating systems is presented below.

- **Linux**

  Linux is a free *Unix-like* operating system, originally created by Linus Torvalds and afterwards developed with the contribution of many programmers all around the world. It includes a really multitasking environment, multi-user capabilities, virtual memory support, shared libraries, demand loading, TCP/IP networking and many more features that justify the title "*Unix-like*". Linux's license is GPL, which means that its source code is available to the public.
  Linux requires at least 4 Megabytes of RAM, and 8 Megabytes if one wants to use X-Windows. It is recommended to have at least 32 Megabytes. It is possible to set a Linux system on a 50 Megabytes hard disk, but 500 Megabytes are preferable. Linux is supported by all X86 processors, but the minimum required CPU is 80386SX. It is also supported by other platforms (e.g. AMD, Compaq Alpha AXP, PowerPC, Sun SPAR and UltraSPARC etc).

- **Mac OS X**

  Mac OS X is a line of proprietary, graphical operating systems developed, marketed, and sold by Apple Inc., the latest of which is pre-loaded on all currently shipping Macintosh computers. It is the successor to the original Mac OS, which had been Apple's primary operating system since 1984. Unlike its predecessor, Mac OS X is a Unix-like operating system built on technology that had been developed at NeXT through the second half of the 1980s and up until Apple purchased the company in early 1997.

  The operating system was first released in 1999 as Mac OS X Server 1.0. Since then, four more distinct "end-user" and "server" editions of have been released, the most recent being Mac OS X v10.4, which was first made available in April 2005.

  Mac OS X used to support the Java Platform as a "first class citizen" — in practice this means that applications written in Java fit as neatly into the operating system as possible while still being cross-platform, and that graphical user interfaces written in Swing look almost exactly like native Cocoa interfaces. Traditionally, Cocoa programs have been mostly written in Objective-C, with Java as an alternative. However, on July 11, 2005, Apple announced that "features added to Cocoa in Mac OS X versions later than 10.4 will not be added to the Cocoa-Java programming interface."

- **Microsoft Windows**

Windows XP is a line of proprietary operating systems developed by Microsoft for use on general-purpose computer systems, including home and business desktops, notebook computers, and media centers. Windows XP is the successor to both Windows 2000 and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT kernel and architecture. Windows XP was first released on October 25, 2001, and over 400 million copies are in use, according to a January 2006 estimate. It is succeeded by Windows Vista, which was released to volume license customers on November 8, 2006, and worldwide to the general public on January 30, 2007.
The most common editions of the operating system are Windows XP Home Edition, which is targeted at home users, and Windows XP Professional, which has additional features such as support for Windows Server domains and dual processors, and is targeted at power users and business clients. Two separate 64-bit versions of Windows XP were also released, Windows XP 64-bit Edition for IA-64 (Itanium) processors and Windows XP Professional x64 Edition for x86-64 processors.
Windows XP is known for its improved stability and efficiency over previous versions of Microsoft Windows.
To install Windows XP Professional, a computer should meet the following minimum hardware requirements:

a) 233 megahertz (MHz) Pentium or higher microprocessor (or equivalent)
b) 128 megabytes (MB) recommended (64 MB of RAM minimum 4 gigabytes (GB) of RAM maximum)
c) 1.5 GB of free space on your hard disk

- ***Operating System Comparison***

This section comprises a short comparison between the different operating systems that were previously introduced.

a) Reasons for using Linux

Linux is cheaper, faster and much more flexible than Windows XP (or other versions). Maintaining a Linux system costs less than a WinNT one (this advantage is also applies to every other Unix system). Linux is the cheapest Unix system. Users can maintain it quickly and easily from any place over the Internet. It supports more devices than FreeBSD, SCO and Solaris/X86.

Because of the fact that its source code is available, all the bugs are fixed in a short time.
From a switching standpoint, the real beauty of Linux is that it runs on the hardware the user already owns, and can be, with some help, made to work with Windows data and, increasingly, even with certain Windows applications

b) Reasons for using Mac

Apple Mac OS X 10.4 is the strongest OS X release yet and a worthy competitor to Windows XP. It performs well, looks great, and offers many modern OS features. It builds on the rock-solid foundation of previous OS X releases, adds a few major new features, and applies a nice spit polish to hundreds of other small features. While it may lack some of the niceties that make Windows more appealing to new users, it does reward those with existing computer skills with a minimalist yet elegant user interface.

c) Reasons for using Windows

Windows XP Professional is built on the core software code used in Windows 2000 and Windows NT Workstation. This code, known as the NT kernel, makes Windows XP more powerful, secure, and stable than Windows Me, Windows 98, or Windows 95. ***Windows XP* is still the most complete desktop environment available for the x 86 platfo*rms. No other OS distribution can match Microsoft XP's ability to support such a wide variety of applications or its very large application base or well-known and easy-to-use user interface***

There is no definite "winner" in this debate, but it suffices to know that all systems are capable of operating adequately well in the HEARTFAID context, and providing support for both hardware and software features.

6.6.1.2 The Java Programming Language

The Java programming language and environment is designed to solve a number of problems in modern programming practice. It started as a part of a larger project to develop advanced software for consumer electronics. These devices are small, reliable, portable, distributed, real-time embedded systems. When the project started the development team intended to use C++, but encountered a number of problems. Initially these were just compiler technology problems, but as time passed more problems emerged that were best solved by changing the language. This gave the incentive for the emergence of Java: A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language. Java encapsulates a number of important concepts, and exposes them to form a balanced, versatile programming environment. Its key features include:

- **Simplicity**

Java was developed by taking the best points from other programming languages, primarily C and C++. Java therefore utilises algorithms and methodologies that are already proven. Java omits many rarely used, poorly understood, confusing features of C++ that admittedly bring more grief than benefit. These omitted features primarily consist of operator overloading (although the Java language does have method overloading), multiple inheritance, and extensive automatic coercions. Error prone tasks such as pointers and memory management have either been eliminated or are handled by the Java environment automatically rather than by the programmer. By virtue of having automatic garbage collection (periodic freeing of memory not being referenced) the Java language not only makes the programming task easier, it also dramatically cuts down on bugs.

Another aspect of being simple is being small. One of the goals of Java is to enable the construction of software that can run stand-alone in small machines. The Java interpreter and standard libraries have a small footprint. A small size is important for use in embedded systems and so Java can be easily downloaded over the net.

Since Java is primarily a derivative of C++ which most programmers are conversant with, it implies that Java has a familiar feel rendering it easy to use.

- **Object-Oriented Design**

Even though Java has the look and feel of C++, it is a wholly independent language which has been designed to be object-oriented from the ground up. In object-oriented programming (OOP), data is treated as objects to which methods are applied. Object-oriented design is very powerful because it facilitates the clean definition of interfaces and makes it possible to provide reusable "software components." Simply stated, object-oriented design is a technique that focuses design on the data (=objects) and on the interfaces to it. Furthermore, it is the mechanism for defining how modules "plug and play".

- **Network-Savvy**

Java has an extensive library of routines for coping easily with TCP/IP protocols like HTTP and FTP. This makes creating network connections much easier than in other OO programming languages. Java applications can open and access objects across the net via URLs with the same ease that programmers are used to when accessing a local file system.

- **Robustness**

Java is intended for writing programs that must be reliable in a variety of ways. Java puts a lot of emphasis on early checking for possible problems, later dynamic (runtime) checking, and eliminating situations that are error prone.

One of the advantages of a strongly typed language (like C++) is that it allows extensive compile-time checking so bugs can be found early. Unfortunately, C++ inherits a number of loopholes in compile-time checking from C, which is relatively lax (particularly method/procedure declarations). In Java, we require declarations and do not support C-style implicit declarations.

The linker understands the type system and repeats many of the type checks done by the compiler to guard against version mismatch problems.
The single biggest difference between Java and C/C++ is that Java has a pointer model that eliminates the possibility of overwriting memory and corrupting data. Instead of pointer arithmetic, Java has true arrays. This allows subscript checking to be performed. In addition, it is not possible to turn an arbitrary integer into a pointer by casting.

- **Security**

Java is intended for use in networked/distributed environments. Toward that end, a lot of emphasis has been placed on security. Java enables the construction of virus-free, tamper-free systems. The authentication techniques are based on public-key encryption.
There is a strong interplay between "robust" and "secure." For example, the changes to the semantics of pointers make it impossible for applications to forge access to data structures or to access private data in objects that they do not have access to. This closes the door on most activities of viruses.

- **Architecture Neutrality and Portability**

Java was designed to support applications on networks. In general, networks are composed of a variety of systems with a variety of CPU and operating system architectures. To enable a Java application to execute anywhere on the network, the compiler generates an architecture-neutral object file format--the compiled code is executable on many processors, given the presence of the Java runtime system.
Being architecture neutral is a big chunk of being portable, but there's more to it than that. Unlike C and C++, there are no "implementation dependent" aspects of the specification. The sizes of the primitive data types are specified, as is the behaviour of arithmetic on them. For example, "int" always means a signed two's complement 32 bit integer, and "float" always means a 32-bit IEEE 754 floating point number. Making these choices is feasible in this day and age because essentially all interesting CPUs share these characteristics.
The libraries that are a part of the system define portable interfaces. For example, there is an abstract Window class and implementations of it for Unix, Windows NT/95, and the Macintosh.
The Java system itself is quite portable. The compiler is written in Java and the runtime is written in ANSI C with a clean portability boundary. The portability boundary is essentially a POSIX subset.

Although the list of Java’s qualities is not exhausted here, the list compiled so far makes a convincing argument for the use of this language. Besides the excellent design and implementation contributed in the development of Java, a large number of APIs have been built upon it, making Java a strong player in almost any type of application***. The Home and Clinical scenarios that are examined in this section call for interaction with either wired or wireless sensors, via serial port or Bluetooth, and XML manipulation. Java I/O is one of the most feature-***

rich implementation areas of the language, and XML manipulation has also received a fair share of attention. The result is a full set of APIs that tackle each of these tasks; the following subsections give a brief presentation of the available choices.

Serial Port I/O

There are many APIs and communication options to allow ***access from a Java application to an RS232 connected device***. Since an important consideration is to find a safe, robust and intuitive interface, two among the multitude of offered packages have been deemed adequate. These are:

- **Java Communications 3.0 by Sun**

The Communications API is a Java extension that facilitates developing platform-independent communications applications for technologies such as Smart Cards, embedded systems, and point-of-sale devices, financial services devices, fax, modems, display terminals, and robotic equipment. The Communications API (also known as javax.comm) provides applications access to RS-232 hardware (serial ports) and limited access to IEEE-1284 (parallel ports), SPP mode.
Implementations of the API are currently available for Solaris SPARC, Solaris x86, and Linux x86.
API serial features:

- Enumeration of ports (administrator and user configurable port mapping)
- Port configuration (baud rate, speed, stop bits, parity)
- Access to EIA232 standard DTR, CD, CTS, RTS and DSR signals
- Transfer of data over RS-232 ports
- Hardware and software flow-control options
- Receive-buffer threshold control
- Asynchronous event option for notification of:
    - Data available on an RS-232 port
    - Port hardware line level changes
    - Port ownership changes within a single JVM

- **SerialPort by SerialIO**

SerialPort is the first published, commercial Java product for serial ports that provides flexible control of serial ports from a Java application. SerialPort is a high-performance class that also provides low-level serial port control. If a device has an RS232 interface, it is certain that one can use SerialPort to communicate with it.
SerialPort has even preceded Sun Microsystems in developing and releasing the *first beta* of javax.comm.SerialPort. The SerialPort package also provides javax.comm.SerialPort on more platforms than any company in the world. One can either use the SerialPort API directly, or use javax.comm.SerialPort, as defined in Sun's specification.

SerialPort allows the Java application to have full control of serial ports. Some additional features are:

- Multi-port cards - 96 ports (can be extended with Enterprise license)
- Bean for serial ports
- Create EXEs for Win32
- Speeds from 110 bps to 460.8 Kbps
- Full handshake protocol, word size, parity, stop bit control
- Control size of send and receive buffers
- Line status checking for CD, CTS, DSR
- Control for setting line state on DTR, RTS
- Send bytes, packets (byte arrays), and strings
- Error reporting including UART overrun, framing, and parity errors
- Ring signal reporting
- Hardware and Software flow control selection
- High-performance byte queue
- Modem utility classes
- Power PCMCIA and CF card power management features

Bluetooth

While Bluetooth hardware has advanced, ***there has been no standardized way to develop Bluetooth applications - until JSR 82 came into play***. It is the first open, non-proprietary standard for developing Bluetooth applications using the Java programming language. It hides the complexity of the Bluetooth protocol stack behind a set of Java APIs that allow you to focus on application development rather than the low-level details of Bluetooth. JSR 82 is based on version 1.1 of the Bluetooth Specification. JSR 82 consists of two optional packages: the core Bluetooth API and the Object Exchange (OBEX) API. The latter is transport-independent and can be used without the former.
The API is intended to provide the following capabilities:

- Register services
- Discover devices and services
- Establish RFCOMM, L2CAP, and OBEX connections between devices
- Using those connections, send and receive data (voice communication not supported)
- Manage and control the communication connections
- Provide security for these activities

JSR 82 requires that the Bluetooth stack underlying a JSR 82 implementation be qualified for the Generic Access Profile, the Service Discovery Application Profile, and the Serial Port Profile. The stack must also provide access to its Service Discovery Protocol, and to the RFCOMM and L2CAP layers.
The APIs are designed in such a way that developers can use the Java programming language to build new Bluetooth profiles on top of this API as long as the core layer specification does not change. To promote this flexibility and

extensibility, the specification is not restricted to APIs that implement Bluetooth profiles. JSR 82 includes APIs for OBEX and L2CAP so that future Bluetooth profiles can be implemented in Java, and these are already being used for that purpose. Figure 9 shows where the APIs defined in this specification fit in a CLDC/MIDP architecture.

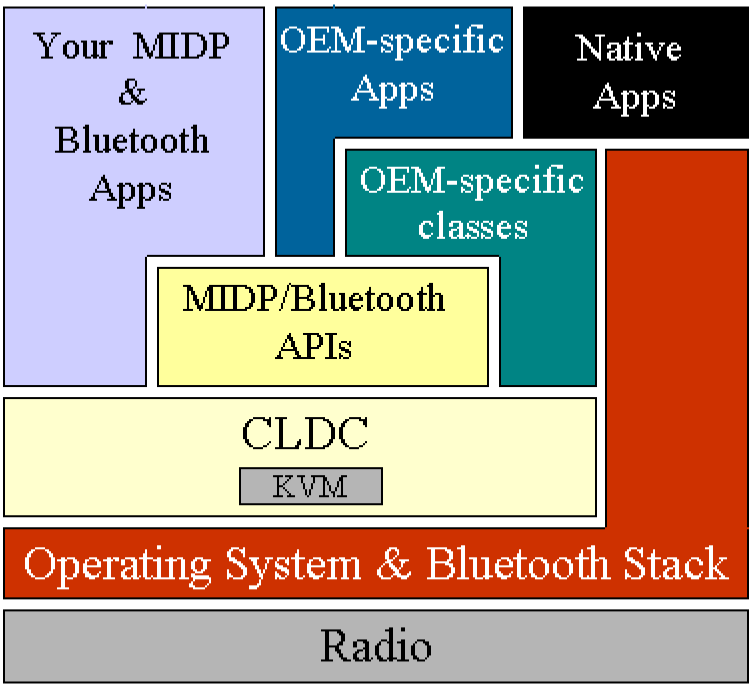


Figure 9 High-level Architecture of J2ME CDLC/MIDP and Bluetooth

XML Processing

In recent years, XML has grown into the most popular data exchange format, hence the demand for efficient XML processing frameworks has also exploded. Sun answered this request by releasing JAXP, the Java API for XML Processing, which enables applications to parse and transform XML documents independent of a particular XML processing implementation. Independent development teams, mainly members of the Open Source community, have specified and implemented competing XML processing standards, which are allegedly more intuitive and flexible. The net effect of these efforts is that a variety of mature and robust Java implementations exist for XML manipulation, so the task of handling and transmitting XML data is greatly simplified. The choice of XML processing engine will be made among the following alternatives.

- **Java API for XML Processing by Sun**

  JAXP leverages the parser standards Simple API for XML Parsing (SAX) and Document Object Model (DOM) so that one can choose to parse data either as a stream of events or by building an object representation of it. JAXP also supports the Extensible Stylesheet Language Transformations (XSLT) standard, giving control over the presentation of the data and enabling data

conversion to other XML documents or to other formats, such as HTML. JAXP also provides namespace support, so one can work with DTDs that might otherwise have naming conflicts.
Designed to be flexible, JAXP allows the implementer to use any XML-compliant parser from within the application. It does this with what is called a *pluggability layer*, which lets developers plug in an implementation of the SAX or DOM API. The pluggability layer also allows developers to plug in an XSL processor, letting them control how the XML data is displayed.

- **Xerces2 by Apache**

Xerces2 is the next generation of high performance, fully compliant XML parsers in the Apache Xerces family. The latest version of Xerces introduces the Xerces Native Interface (XNI), a complete framework for building parser components and configurations that is extremely modular and easy to program.
The Apache Xerces2 parser is the reference implementation of XNI but other parser components, configurations, and parsers can be written using the Xerces Native Interface.
Xerces2 is a fully conforming XML Schema processor. Furthermore, Xerces2 provides a complete implementation of the Document Object Model Level 3 Core and Load/Save W3C Recommendations and provides a complete implementation of the XML Inclusions (XInclude) W3C Recommendation. It also provides support for OASIS XML Catalogs v1.1.
Xerces2 is able to parse documents written according to the XML 1.1 Recommendation, except that it does not yet provide an option to enable normalization checking as described in section 2.13 of this specification. It also handles namespaces according to the XML Namespaces 1.1 Recommendation, and will correctly serialize XML 1.1 documents if the DOM level 3 load/save APIs are in use.

- **JDOM**

JDOM is an open source, tree-based, pure Java API for parsing, creating, manipulating, and serializing XML documents. Although it's similar to the World Wide Web Consortium's (W3C) DOM, it's an alternative document object model that was not built on DOM or modelled after DOM. The main difference is that while DOM was created to be language-neutral and initially used for JavaScript manipulation of HTML pages, JDOM was created to be Java-specific and thereby take advantage of Java's features, including method overloading, collections, reflection, and familiar programming idioms.
Like DOM, JDOM represents an XML document as a tree composed of elements, attributes, comments, processing instructions, text nodes, CDATA sections, and so forth. The entire tree is available at any time. Unlike SAX, JDOM can access any part of the tree at any time. Unlike DOM, all the different kinds of nodes in the tree are represented by concrete classes rather than interfaces. JDOM also offers support for XML Namespaces, XSLT, and XPath.

- **dom4J**

dom4j is an Open Source XML framework for Java that allows an application to read, write, navigate, create and modify XML documents. It was initially developed as a fork of JDOM, although its architectural design has little in common with the latter. It combines features of the Document Object Model (DOM) and Simple Application Programming Interface for XML (SAX), and includes support for XML Path Language (XPath), Java 2 Collections, Java API for XML Parsing (JAXP), Transformation API for XML (TRaX) and Extensible Style Language Transformations (XSLT).
dom4j is distributed under an open source, Apache-style license that does not restrict users to creation of open source products only.

6.6.2 Wireless devices (Home, Clinical Scenario and On-The-Move scenarios)

As stated before, one of the major requirements while designing and implementing the Data Acquisition and Transmission Infrastructure is to use for data collection devices that the user already uses in his everyday life. In this scope, mobile smart phones were considered as the strongest candidate solution. In the last years we have seen an increased growth in the smart phone marketplace. Manufacturers are selling more and more high-end devices and the prospects are that the percentage these devices have of the whole handset market will increase over the coming years. Smart phones enable a possibility for enhanced services and applications due to the more powerful operating system they use; hence their use as data colleciton devices in the healthcare domain is more and more considered as an approach.

The operating system is one of the most strategic parts in this context and it is important for the operators to identify the possibilities it enables. Today, we have competing operating systems, which all try to become the de facto standard, but what will the market look like in a couple of years? What are the pros and cons of the different systems and how will these affect the operators' businesses?

In the pre-GSM era of mobile phones there were several mobile telephony systems in the world: Nordic Mobile Telephony (NMT), Advanced Mobile Phone Service (AMPS), Total Access Communications System (TACS), to mention a few of the most important ones. This was the first generation, or the analog era, of mobile phones. These were all incompatible systems, and basically all countries used their own system. Consequently, when the work with the second generation system (2G) was started, in the beginning of the 1980's, one of the main goals was to unite all European countries under the new digital telephony system (GSM). Later, GSM became the global standard for mobile telecommunications. These mobile systems did not include as many demanding features as the enhanced second generation (2.5G) and now third generation (3G) of mobile phones do, so all manufacturers used their own software platforms in their phones.

As the industry developed even more sophisticated mobile devices there was a need to have a real operating system, which could offer a better platform for adding new features and complexity. Another aspect is that mobile phones constantly are coming closer to the Internet and what it has to offer. Moreover, to provide the compatibility between mobile phones from different manufacturers it was crucial to have a common operating system. So what would this operating system be?

Big players in the mobile phones industry decided to make their own system to be used as the platform for high-end smart phones. One major reason for this was to hinder Microsoft from taking over the market with a mobile phone version of Windows, but also to be able to control the development of the platform and maintain its suitability for mobile phones. This resulted in an alliance named Symbian.

Other players, in fields close to the mobile phone industry, are manufacturers making other handheld devices like Personal Digital Assistants (PDA's), and also operating systems for other types of devices. Adding these to the equation we will get the main players, which are the targets for analysis in this paper: Symbian, Palm, Microsoft and Linux.

In the following paragraphs, we will present the current main players competing in the domain of operating system for mobile devices, and, analyze the systems they offer from a technical and a business perspective. New services enabled by the smart phones are presented in order to see the possibilities available and how these will drive the operators' business opportunities. In addition, some ideas, which vendor will win the battle of mobile operating system, are discussed, and, finally conclusions are drawn.

6.6.2.1 Competing Mobile Operating Systems

In the following sections we will present the main rivaling operating systems used in mobile devices: Symbian, Palm OS, Windows Mobile and Linux. Each system has a different history and primary target market segment, but now as devices are more similar and the marketplaces start to intersect we get a new mixture of players and interests. In order to better understand the advantages and drawbacks each system offers, we need to compare them, both from a technical and a business viewpoint.

Symbian

Symbian was established as an independent company by Nokia, Ericsson, Motorola and Psion in 1998. It is designed for the specific requirements of open, data-enabled 2G, 2.5G and 3G mobile phones. The main characteristics are integrated multimode mobile telephony, open application environment, open standards and interoperability, multi-tasking, fully object-orientation, flexible user interface design and robustness.

Nokia is the biggest owner in Symbian and has taken the role as the real forerunner of this technology. Since ***Symbian is the main operating system for smart phones*** and also has all giants in the industry backing it up, it is targeted to

fully support these devices. The operating system is 32 bit and it provides all current technologies in mobile phones such as GSM, GPRS, WAP, SMS, MMS, E-mail, Bluetooth et cetera. In the newest versions the focus has been on security, graphics, real time applications, and device management for the operators. Because Symbian develops alongside the mobile phone industry, features are added to support new innovations and technologies, and this is very important when considering the competitiveness of the operating system.
Currently, there are 14 mobile phone manufacturers licensing Symbian: Arima, BenQ, Fujitsu, Panasonic, Levono, LG Electronics, Motorola, Nokia, Samsung, Sanyo, Sendo, Sharp, Siemens, and Sony Ericsson. In addition, Mitsubishi is negotiating for a license. These include all the big players in the mobile phone industry, so the support for this platform is enormous.
Symbian is closely related with the C++ programming language, so potential developers are many. The community supporting it is also well developed and numerous companies are supplying solutions for Symbian.

Windows Mobile

The biggest player in operating systems, Microsoft, has also entered the smart phone operating system's market with its Windows Mobile operating system (Windows Mobile 2005). The target segment for this OS is all kinds of mobile devices, like PDA's and mobile phones.
It all started in the beginning of the 1990's with Windows CE, which was aimed at handheld PC's. This product line has later been adapted for smart phones and other mobile devices. In order for Microsoft to be able to penetrate the mobile phone market contracts with several mobile device manufacturers were signed. For example, Motorola and Samsung are also making mobile phones based on Windows Mobile. In addition to these, there are numerous smaller smart phone manufacturers, which only use Windows Mobile as the operating system.
Windows Mobile is a scaled down version designed for smaller devices with less resources than normal PC's. However, it is built based on Windows CE kernel, so the main key features come from compatibility with other Windows systems. It is 32 bit and its main features are it multi-threading nature, rich and powerful applications, well-known and easy-to-use user interface, compatibility with other Windows's and robustness.
The real power of Windows Mobile is the compatibility with other Windows's, which means developers can easily port applications to this new scaled-down version, and, users are familiar with the interface and how to use the applications. Another very important issue is the huge community of developers developing in the Windows environment. These are the real value-adders, which boost the sales of new Windows operating systems.
Windows Mobile is maybe not focusing on telecommunication-specific technologies that much, even if it supports the main technologies like GSM, CDMA, SMS, MMS, et cetera. It focuses on trying to integrate mobile phones with the mother operating system: Windows.

Palm OS

PalmSource is the maker of Palm OS, which is designed and developed for mobile devices, especially PDA's. The operating system was originally developed for a PDA manufactured by US Robotics, but later it has changed its name from Pilot to Palm OS.

Palm OS is a 32 bit scalable and modular operating system which based on standards, has a multi-threaded nature, supports an extensible multimedia framework as well as a wide application range.

An important driver for Palm OS is the huge community and the large applications offerings: more than 400,000 developers and 20,000 (PalmSource 2005).

MontaVista Linux

Apart from the commercial operating systems for smart phones there are Open Source alternatives based on Linux. MontaVista Software Inc. offers a modified version of Linux, MontaVista Linux, made specifically for embedded devices. Mobile phones are just one part of the segment. The company was founded in 1999 and it is located in Silicon Valley in California, USA. It is privately held company funded by investors like Alloy Ventures, IBM, NTT DoCoMo among others.

There are several important key features, which make Linux an interesting choice when looking for an operating system:

- open-source,
- flexibility and choice,
- scalable,
- performance,
- robust,
- multi-threaded,
- memory protection,
- portable.

The most valuable property is open-source, which means that it can be modified freely and customized for a specific manufacturer's products.

Currently, there are not so many mobile phone manufacturers using Linux in their products, but it is prospected (LinuxDevices Feb. 23, 2005) to change in the coming years. Nevertheless, Motorola, NEC, Panasonic and Samsung are big mobile phone manufacturers offering phones running on Linux.

The Open Source movement is really something important today, and the community supporting it and Linux is large and widely spread. This adds value to using Linux as the platform on mobile devices.

Other players

Smaller upcoming operating systems trying to challenge the current market leaders are for example SavaJe and Radixs MXI.

SavaJe is an operating system based on Java, which has some interesting features like based on open standards, highly customizable and full Java Second Edition (J2SE) API compatible. SavaJe was founded in 1999 by members from the Lucent Technologies Inferno operating system team.

Radixs was also founded in 1999 in Singapore. After several years of research it finally launched its operating system Motion eXperience Interface in September 2003. The idea behind the operating system is to be able to run any application, Windows, Linux, or Palm OS on the system.

In addition to these, there are some other smaller and also legacy operating systems. We will not address either of these in this paper.

Comparison

The size of the mobile phone marketplace and the potential of the smart phone marketplace are too big for any major operating system developer to ignore, and this is the reason for such diverse companies competing in it. Windows and Linux both come from the PC world, while Palm OS has been the main player in the PDA operating system marketplace. Symbian, on the other hand, being an alliance between the major cellular phone manufacturers has a great advantage due to this. Currently, the market shares in the mobile device marketplace are dominated by Symbian, Windows Mobile and Palm OS; this can be seen in figure 1 below.

Symbian is the leader in voice-centric devices, while Microsoft has the lead in data-centric devices. This reflects the strategies the vendors have: Symbian, which comes from the telecommunication industry, is focusing mostly on technologies enabled by the networks. It tries to become the standard operating system for these devices. Microsoft on the other hand is everywhere trying to make their other software solutions work in any environment to get synergies between all its solutions. To Linux’ strategy we can directly add openness and a support for a wide range of devices, from supercomputers to small embedded systems.

When only focusing on the technical aspects of the different systems, it is quite evident that all the vendors offer more or less the same basic functionalities. In addition to this, we can identify some issues where they differ. For example, Symbian is made by mobile phone manufacturers to be used in mobile phones, so their focus lies on supporting the mobile phone technologies and to work well in these devices. This makes its market segment narrower than the other competitors’. Palm OS has also been focusing on a specific segment, the PDA’s, but in recent years they have found out that they need to enter the mobile phone marketplace as well. This has been done by extending the own system to work in these devices, and also through the acquisition of China MobileSoft. Microsoft on the other hand has a huge selection of operating systems to be used on almost any architecture – this is both good and bad. The advantages come from know-how in the field, and huge resources to be allocated for developing an operating system. The other side of the coin adds complexity to the operating system due to integration with the other Windows’s, and also the fact that Windows is not specifically tailored for the needs found in small, handheld mobile devices. Then we have the underdog Linux, which is fighting on all levels in the operating systems marketplace. It has very good key features, but the lack of control might hinder it from getting real big market shares. However, there are a lot of people, who believe in Linux and some even think it will be the future market leader (The Standard: Interview with PalmSource CEO David Nagel). Another very important issue, for many countries and organizations, is that Linux is open-source, which means you are not using something you cannot analyze and see what it actually does.

Concentrating only on smart phones, ***Symbian is very interesting, because of the fact that it addresses the most important issues found in these devices***. The resources and low, but still there is a great need to enable the very latest technology to be used by application developers. Linux key aspect is the high level of customization, which might be required in order to fulfill all the needs an operator sets. Palm OS is a turn-key solution, which feels a little bit too inflexible and the market shares are also going down, taken by Microsoft. Windows Mobile is quite heavy and is not optimized for devices with few resources and the development platform is also maybe a little bit too unfocused on the smaller devices.

Which Operating System Wins?

Currently, Symbian is the clear leader, but Microsoft is taking market shares every year, mostly from Palm. In addition, Microsoft's large capital resources can also promote and boost the sales of its software. This is a great resource, which no other player currently possesses.

While there is no dominant design, manufacturers offer several different systems throughout their product line. Nokia is, however, only focusing on Symbian-based phones, which can be motivated by being the biggest owner in Symbian. Is this a good thing? Retailers and operators on the other hand want to differentiate and offer several different platforms. Is this only due to no clear winner or because customers need different platforms?

The outcome of this battle depends a lot on the mobile phone manufacturers: Nokia and Sony Ericsson are pushing Symbian, new manufacturers in Asia are using Linux and also Windows Mobile to some extent, and we have Palm in the PDA business in the US. Microsoft's important role in the overall software business can also be seen when Nokia licensed the Microsoft Exchange Server ActiveSync protocol to be used in mobile devices. This shows how important compatibility with Microsoft software is. But, the question is if this will determine a winner?

One thing, which has not been addressed, is security. It will be even more important as we get devices online and later all-IP networks. Symbian and Linux are both focusing very much on this issue, and Microsoft's reputation is not the best in this area.

The mobile phone business is constantly emerging markets dominated by other devices, and this result in a lot of really big players to count on. In order to determine a clear winner we need a revolution or a really superior design, and this can not be seen right now. Therefore, we will most probably still have different systems and several players competing for the ultimate victory. The compatibility between the devices could be reached by licensing technology and communicating using common protocols. Geographically, we will see the same tendency as we see today: the major players' home markets will be dominated by their design and offerings.

6.6.2.2 The role of Java programming Language in the development of wireless applications

When a mobile device is considered as an acquisition device for biomedical data, the next logical question to be answered is: which programming language will be used in order to develop the code that **enables the low-level communication with the sensors, possibly enhances the acquired biomedical data and also transmits the data to the HEARTFAID platform**. Considering the aforementioned requirements, J2ME is the most important approach, compared to the development of native code, because of its portability, ease of programming and great support of APIs for e.g. Bluetooth Connectivity, IP connectivity, XML parsing etc. For this reason, in the following paragraphs we give an overview of the features of J2ME related to the requirements a mobile data collection device should comply with, in the scope f the HEARTFAID project.

J2ME

J2ME is a collection of Java APIs for the development of software for resource-constrained devices such as PDAs, cell phones and other consumer appliances. Java ME is formally a specification, although the term is frequently used to also refer to the runtime implementations of the specification. Java ME was developed under the Java Community Process as JSR 68. The evolution of the platform has abandoned the umbrella Java Specification Request in favor of separate JSRs for the different flavors of Java ME.

Java ME was designed by Sun Microsystems and is a replacement for a similar technology, PersonalJava. Sun only provides a reference implementation and most work targeting a non-Intel-based small device will require a vendor-supplied JVM to be available on the device.

Java ME has become a popular option for creating applications for cell phones, as they can be emulated on a PC during the development stage and easily uploaded to the phone. Sun Microsystems has tended not to provide free binary implementations of its Java ME runtime environment for mobile devices, rather relying on third parties to provide their own, in stark contrast to the numerous binary implementations it provides for the full Java platform standard on server and workstation machines.

J2ME Configurations and Profiles

Java ME devices implement a *profile*. The most common of these are the Mobile Information Device Profile aimed at mobile devices, such as cell phones, and the Personal Profile aimed at consumer products and embedded devices like Set-top boxes and PDAs.
A profile is a superset of a *configuration*, of which there are currently two: Connected Limited Device Configuration and Connected Device Configuration.

a) Connected Limited Device Configuration
The CLDC contains a strict subset of the Java class libraries, and is the minimal needed for a Java virtual machine to operate. CLDC is basically used to classify myriad devices into a fixed configuration.

- **Mobile Information Device Profile**: Designed for cell phones, MIDP boasts an LCD-oriented GUI API, and MIDP 2.0 includes a basic 2D gaming API. Applications written for this profile are called MIDlets. Almost all new cell phones come with a MIDP implementation, and it is now the de facto standard for downloadable cell phone games. However, many cellphones can run only those MIDlets that have been approved by the carrier, especially in North America.
- **Information Module Profile:** The Information Module Profile (**IMP**) is a Java ME profile for embedded, "headless" devices such as vending machines, industrial embedded applications, security systems, and similar devices with either simple or no display and with some limited network connectivity. Originally introduced by Siemens Mobile and Nokia as JSR-195, IMP 1.0 is a strict subset of MIDP 1.0 except that it doesn't include user interface APIs — in other words, it doesn't include support for the Java package `javax.microedition.lcdui`. JSR-228, also known as IMP-NG, is IMP's next generation that is based on MIDP 2.0, leveraging MIDP 2.0's new security and networking types and APIs, and other APIs such as `PushRegistry` and `platformRequest()`, but again it doesn't include UI APIs, nor the game API. IMP applications are called *IMlets*, but in reality they are MIDlets. They subclass MIDlet, and follow the same packaging, deployment, security and life-cycle as MIDlets.

b) Connected Device Configuration
CDC is a subset of Java SE, containing almost all the libraries that are not GUI related. It is richer than CLDC.

- Foundation Profile: A headless version of Java SE.
- Personal Basis Profile: Extends the Foundation Profile to include lightweight GUI support in the form of an AWT subset.
- Personal Profile: This extension of Personal Basis Profile includes a more comprehensive AWT subset and adds applet support.

J2ME Optional Packages

An optional package is also a set of APIs, but unlike a profile, it does not define a complete application environment. An optional package is always used in conjunction with a configuration or a profile. It extends the runtime environment to support device capabilities that are not universal enough to be defined as part of a profile or that need to be shared by different profiles.

Consider the Wireless Messaging API (WMA), a set of classes for sending and receiving Short Message Service (SMS) messages. Because the WMA is an optional package, it can be included on any J2ME device with SMS capabilities, not just MIDP-enabled cellphones. If WMA were part of a specific profile, such as MIDP, its use would have been limited to that profile and its supersets.

Because, just like profiles and configurations, optional packages are specified through the Java Community Process, each has its own reference implementation (RI) and test compatibility toolkit (TCK). Besides aiding the vendors in implementing optional packages as part of their runtime environments, the RI and TCK also ensure that that implementation are done consistently and correctly, no matter which device is being used.

It is the vendor of a Java runtime environment – in many cases, the device manufacturer – that controls which optional packages are available, just as it controls which configuration is used and which profiles are available. In general, it's not possible to use an optional package if it's not preloaded onto a device as part of the runtime environment. It takes some time after an optional package's specification has been finalized before implementations of the package appear in commercially available devices. The Mobility Optional Packages are:

- Information Module Profile (IMP), JSR 195
- Wireless Messaging API (WMA); JSR 120, JSR 205
- Mobile Media API (MMAPI), JSR 135
- Location API for J2ME, JSR 179
- SIP API for J2ME, JSR 180
- Security and Trust Services API for J2ME, JSR 177
- Mobile 3D Graphics, JSR 184
- J2ME Web Services APIs (WSA), JSR 172
- ***Bluetooth API, JSR 82***
- PIM and FileConnection API, JSR 75
- J2ME RMI, JSR 66
- JDBC for CDC/Foundation Profile API, JSR 169

In the case of the HEARTFAID project, where the acquisition of the medical data from the sensors will be based on Bluetooth, the existence of Bluetooth API in the target devices that will acquire the data is considered mandatory.

XML in J2ME

More and more enterprise and Java technology projects are making use of XML as a medium to store data in portable fashion. But due to the increased processing power demanded by XML parsers, J2ME applications have largely been left out of this trend. Now, however, ***small-footprint XML parsers for Java are emerging, that will allow MIDP programmers to take advantage of the power of XML***.

- **XML parsing in a MIDP environment**

XML parsers can be used in J2ME applications to interface with an existing XML service. XML parsers tend to be bulky, with heavy run time memory requirements. In order to adapt to the MIDP environment, XML parsers must be small to meet the resource constraints of MIDP-based devices. They should also be easily portable, with minimum effort required to port them to MIDP.
Two frequently used XML parsers for resource-constrained devices are kXML and NanoXML. kXML is written exclusively for the J2ME platform (CLDC and MIDP). As of version 1.6.8 for MIDP, NanoXML supports DOM parsing

- **Performance issues in deploying XML parsers**

There are several performance issues that have to be examined while deploying XML parsing in a MIDP application:

- *Increase in size:* An XML parser is code-intensive and increases the overall size of an application. This is a particularly important consideration for resource-constrained MIDP devices. There are several optimization techniques in order to fight code expansion. First, resource files that are not in use have to be removed. Also obfuscators have to be used, that will remove unused classes, unused methods, and variables from the JAD file.
- *Heavy string parsing:* XML parsers use intensive string parsing to perform their jobs; this will add to the overhead in MIDP applications with low runtime memory. XML documents that J2ME applications parse need to be small and contain as much useful information as possible.
- *Slow response time:* As the MIDP application parses a relatively large amount of XML data, the response time will increase. The XML files to be parsed should be small, and the parsing should be done in a thread of execution that is separate from the main application.

Taking into account the above performance issues, although an considering that an XML messaging approach will be adopted for the transmission of biomedical data to the HEARTFAID platform, although the selected smart phones may have the ability to pack the acquired data into XML messages and transmit them, this procedure involves the transmission of large volume of data. Consequently, in HEARTFAID an alternative solution will be followed according to which the collected data are sent to an intermediate gateway, where the XML messages are generated and finally sent to HEARTFAID platform.

6.7 Requirements related to biomedical data transmission

In order to describe the requirements related to biomedical data transmission over public networks, the major parameters which need to be taken into account are the following:

a) ***Connectivity requirements*** related to the nature of the data to be transmitted:

Depending on the nature of the biomedical data to be transmitted, the requirements of the data transmission infrastructure may vary. Generally, the data to be transmitted by the system can be classified as follows, according to their nature:

- Data that need to be transmitted continuously and are mainly related to the real-time monitoring of a patient's vital sign (for example a continuous transmission of a patient's ECG to the platform)
- Data that need to be transmitted at specific times and are mainly related to frequent monitoring of a vital sign during the day (for example a patient measures his blood pressure three times per day and each measurement is sent to the platform when it is performed).

When the transmission of biomedical data involves the real-time monitoring of a person's vital sign (e.g. an ECG monitoring), there is a need for a communication channel that ***supports continuous connectivity*** between the data collection device and the platform. In the second case, continuous connectivity is not crucial, since the device may connect to the platform, upload the measurements and then disconnect.

b) ***Bandwidth requirements*** related to the volume of the data to be transmitted

Another critical parameter related to the data transmission is the volume of data to be transmitted. This parameter directly affects the bandwidth characteristics of the link between the data collection device and the platform. In case of transmission of large volume medical data (e.g. transmission of medical images), it is obvious that there is a need for a high bandwidth communication channel. Of course, the bandwidth characteristics define the ***latency*** relevant to the transmission of data, a parameter which can be considered as critical especially when the transmission of real-time data is desired.

c) ***The environment*** in which the data acquisition and transmission takes place

When the data acquisition takes place in the home environment, the available connectivity, speed and bandwidth conditions are much different than in the case where the data are collected in an outdoor environment.

The following table briefly depicts the transmission requirements in each of the healthcare environments, by grouping the aforementioned parameters.

	Connectivity	Bandwidth	Latency
Clinical Environment	Must be continuous	Must be high	Must be low
Home Environment	Desired continuous	Desired high	Desired low
On-the-move environment	Could be continuous	Could be high	-

According to the previous table we could say that in the clinical environment we envisage the use of medical devices/sensors which produce continuous, large volume and possibly real-time data. This combination imposes the need for a continuous, high bandwidth and low latency connection between the data collection device and the platform. These characteristics could be fulfilled by, for example and ADSL line connecting the data collection device in the hospital with the platform over Internet.

In the case of the home environment, the need for continuous connectivity and high bandwidth is still evident, but since the services / sensors used in this scenario will be mainly homecare appliances (e.g. blood pressure monitors, weighing scales etc), the needs for continuous connectivity, high bandwidth and low latency are not mandatory. However, since current DSL technologies offer these characteristics to home users in reasonable prices, we could say that such characteristics are desired in order to allow for incorporation of devices producing high volume and continuous data in the in-home environment.

Finally, in the on-the-move environment, due to the fact that the transmission of data takes place over a public cellular wireless network, the underlying technology cannot guarantee continuous connectivity (mainly due to possible lack of coverage), although this feature is desired in terms of transmission of data over IP using GPRS/3G networks. Moreover, since generally the data collection devices in this scenario are devices characterised by low computational and storage profile and therefore cannot manipulate large volumes of data, high bandwidth is not a crucial parameter either, although using today's state of the art mobile devices and wireless networks we could enable the use of such applications.

6.7.1 Secure transmission of medical data

A significant issue facing security professionals, especially in healthcare organizations, is the secure transmission of confidential and proprietary information, and protected health information (PHI). When many organizations think of secure transmission, the conversation generally turns to encryption and encrypted e-mail. The main purpose of this section is to explore secure data transmission options that are available to help meet regulatory and legal requirements.

The HIPAA Security Rule, references secure transmission and the use of encryption. Although the Rule does not require the use of encryption, it's included as an "addressable" implementation specification. In other words, a healthcare organization covered under HIPAA has three choices: implement the specification as it appears in the Rule, implement an alternative that is equivalent to the specification or document why the specification is not applicable and therefore is not implemented.

Given the availability and affordability of encryption technology today, it is difficult for a healthcare organization to justify not using some form of it when transmitting PHI. A number of vendors offer a variety of reasonably priced encryption hardware and software, as well as outsourcing options. Now we'll review the options in more detail.

- E-mail encryption

A number of vendors offer products that encrypt e-mail messages, are easy to use and provide the ability to send private data, including e-mail attachments, securely. The recipient can respond using the same encryption method. Many of these products are Web-based. They work by sending a link to the recipient, who then clicks on it and logs on to a secure e-mail server, which the organization either owns or outsources to an appropriate vendor. The recipient is then able to read the e-mail and any attachments securely, and send a secure response including attachments if needed.

There is also non-Web-based technology that allows transportation of secure messages from one person or organization to another, the most common of which is public key infrastructure (PKI). PKI requires an exchange of keys used to unlock the encrypted file. For example, Bob wants to send a secure e-mail to Sue, so he gives her a copy of his public key to open his encrypted message. Bob retains the private key he used to encrypt the message or file, which he can also use, especially with a digital signature, to authenticate himself as the sender. A digital signature is a small electronic file that is unique to each sender and specifically authenticates his or her identity. In many states, a digital signature can be used and is enforceable to the same extent as an original signature on a contract or other legal document.

There haven't been any large PKI deployments as of yet, mainly due to it being cumbersome, and the difficultly of administering and managing keys. However, PKI has been successful with small deployments and is frequently used for sending large files between organizations such as health plans and healthcare clearinghouses.

One method of secure data transmission often used in conjunction with PKI to encrypt and authenticate large data files, is secure file transfer protocol (FTP). However, it is not used for transmission between individuals. The technology is readily available and recommended for organizations transmitting large amounts of data, such as claims transactions and electronic remittance advices through clearinghouses.

- Web site encryption

Organizations that use the Web to collect and transmit sensitive data to customers or other organizations need to secure their Web site. The general standard is the use of secure socket layers (SSL), which encrypts data transmitted via a Web site. Upon opening an Internet browser, an open or closed lock appears in the lower right hand corner of the Web site. If the lock is closed, it means the data transmitted over the Web site is secure, generally by SSL. This allows the transmission and collection of private data over a Web site, without worrying about a hacker accessing it. There is no such thing as security without risks, but the use of SSL and secure Web sites when transmitting data significantly reduces the risk of it being inappropriately intercepted. Secure Web sites can be established by using internal Web analysts/programmers or working with a vendor who has expertise in creating an appealing and secure Web presence.

- Application encryption

Some organizations transmit data between applications, such as an electronic health record. It is wise to view such transmissions, if the data travels outside an organization, as any message sent over the Internet, meaning it's subject to interception and, unless properly protected, misuse. When transmitting sensitive data between applications, it is sound and good security practice to evaluate the encryption capabilities of the application(s) and implement an encryption solution beforehand. An organization can obtain this technology from the vendor that manufactures the application or a custom-programmed product that accommodates application functionality while protecting the data as it travels from one point to another.

- Remote user communication

Remote users present an additional security risk, because they are often communicating between their home and an organization. This means they not only need to be aware of secure data transmission requirements, but also other information security risks associated with remote access to confidential information. To secure communication with remote users, install a virtual private network (VPN), which encrypts all the data sent between its users. This technology is readily available on the market, and it is advisable that organizations with remote users install it. If a VPN is not established and a modem is not in use (which is generally not an efficient method of accessing a company network), all data transmitted over the Internet is subject to interception and inappropriate use.

- Laptops and PDAs

These portable devices can be easily lost or stolen. Therefore, it is wise for organizations using these devices to transport confidential information to encrypt the data stored on those devices. This protects the organization against inappropriate data disclosure if the portable device is lost or stolen. Encryption programs are available for portable devices and the cost of such software is reasonable and affordable, even for smaller organizations.

- Wireless networks

Wireless threats are on the rise and unsecured wireless networks are significant points of vulnerability and open up organizations to easy hacker access. Therefore, it's becoming increasingly important, to prevent access by anyone not authorized to access the network. Also, encrypt all data transmitted between wireless devices to prevent inappropriate disclosure of confidential information. Laptops connected to wireless networks are becoming more common, especially in hospital emergency rooms where medical and health insurance information is collected. These laptops communicate with the organization's wireless server and update applications, health records, etc. This data is generally sensitive and needs the extra layer of protection that encryption provides.

6.7.1.1 Java Cryptography

The Java Cryptography Extension (JCE) is a set of packages that provides a framework and implementations for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms. Support for encryption includes symmetric, asymmetric, block, and stream ciphers. The software also supports secure streams and sealed objects.

JCE was previously an optional package (extension) to the Java™ 2 SDK, Standard Edition (Java 2 SDK), versions 1.2.x and 1.3.x. JCE has now been integrated into the Java 2 SDK, v 1.4.

JCE is based on the same design principles found elsewhere in the JCA: implementation independence and, whenever possible, algorithm independence. It uses the same provider architecture. Providers signed by a trusted entity can be plugged into the JCE framework, and new algorithms can be added seamlessly.

The JCE API covers:

- Symmetric bulk encryption, such as DES, RC2, and IDEA
- Symmetric stream encryption, such as RC4
- Asymmetric encryption, such as RSA
- Password-based encryption (PBE)
- Key Agreement
- Message Authentication Codes (MAC)

The Java 2 SDK, v 1.4 release comes standard with a JCE provider named "`SunJCE`", which comes pre-installed and registered and which supplies the following cryptographic services:

- An implementation of the DES (FIPS PUB 46-1), Triple DES, and Blowfish encryption algorithms in the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Propagating Cipher Block Chaining (PCBC) modes. (Note: Throughout this document, the terms "Triple DES" and "DES-EDE" will be used interchangeably.)
- Key generators for generating keys suitable for the DES, Triple DES, Blowfish, HMAC-MD5, and HMAC-SHA1 algorithms.
- An implementation of the MD5 with DES-CBC password-based encryption (PBE) algorithm defined in PKCS #5.
- "Secret-key factories" providing bi-directional conversions between opaque DES, Triple DES and PBE key objects and transparent representations of their underlying key material.
- An implementation of the Diffie-Hellman key agreement algorithm between two or more parties.
- A Diffie-Hellman key pair generator for generating a pair of public and private values suitable for the Diffie-Hellman algorithm.
- A Diffie-Hellman algorithm parameter generator.
- A Diffie-Hellman "key factory" providing bi-directional conversions between opaque Diffie-Hellman key objects and transparent representations of their underlying key material.
- Algorithm parameter managers for Diffie-Hellman, DES, Triple DES, Blowfish, and PBE parameters.
- An implementation of the HMAC-MD5 and HMAC-SHA1 keyed-hashing algorithms defined in RFC 2104.
- An implementation of the padding scheme described in PKCS #5.
- A keystore implementation for the proprietary keystore type named "JCEKS".

6.7.1.2 The Bouncy Castle Java Crypto API

The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms; it was developed by the Legion of the Bouncy Castle. The package is organised so that it contains a light-weight API suitable for use in any environment (including J2ME) with the additional infrastructure to conform the algorithms to the JCE framework.

Except where otherwise stated, the software is distributed under a license based on the MIT X Consortium license. The OpenPGP library also includes a modified BZIP2 library which is licensed under the Apache Software License, Version 1.1.

The Java **Bouncy Castle Crypto APIs** consist of the following:

- A lightweight cryptography API.
- A provider for the JCE and JCA.
- A clean room implementation of the JCE 1.2.1.
- A library for reading and writing encoded ASN.1 objects.
- A light weight client-side TLS API.

- Generators for Version 1 and Version 3 X.509 certificates, Version 2 CRLs, and PKCS12 files.
- Generators for Version 2 X.509 attribute certificates.
- Generators/Processors for S/MIME and CMS (PKCS7/RFC 3852).
- Generators/Processors for OCSP (RFC 2560).
- Generators/Processors for TSP (RFC 3161).
- Generators/Processors for OpenPGP (RFC 2440).
- A signed jar version suitable for JDK 1.4-1.6 and the Sun JCE.

The lightweight API works with everything from the J2ME to the JDK 1.6.

6.8 Requirements related to data collection and manipulation (Electronic Health Record)

As reported in the DoW of the HEARTFAID project, one of the main objectives of the project will be study and implementation of a HEARTFAID EHR to be adopted in the cardiovascular context. This EHR will be necessary for the traceability, collection and integration of the data identified by the clinical partners as relevant for the project purposes.

The starting point for the design of the HF EHR should have been the single EMRs/EPRs (see Section 5.7.3) actually adopted by clinical partners involved in the HF project; nevertheless, during the early stages of task T3.1, it was ascertained that no suitable solutions are actually in use by the cardiovascular centres where the HF platform will be validated. Therefore, preliminary studies have been started aimed at identifying a suitable tool that could be used within the consortium. After a deep analysis of the needs of the clinical partners, the Consortium decided to extend an existing general purpose EMR developed by Synapsis. This EMR can be easily configured to the specific requirements of the cardiovascular experts and it can be tuned on the specific HF contexts. This way, it will be possible to overcome the lack of existing EMR and the activities of the technical partners will be directed towards the following objectives:

- Definition, adoption and validation of a cardiovascular EMR;
- Definition and implementation of a suitable middleware to support the integration of the new cardiovascular EMR into the EPR solution conceived by Synapsis;
- Move towards the HF EHR.

The technological teams are already working on the first point: in accordance with the clinical partners, the general purpose EMR developed by Synapsis is being adapted to the specific needs of the clinical partners.

A first important requirement for the design and development of an effective HEARTFAID EHR will be the existence on each pilot site of specific *vertical* EMR for the cardiovascular context. In fact, it should be clear that while the EHR is patient centric and should be able to reconstruct the healthcare history of a patient, this can be done on the basis of vertical EMR that provide information of that patient related to specific pathological context (cardiovascular, neurological, etc.).

These EMR should guarantee a full integration both with the EHR and with the Health Information System (HIS) of the healthcare structures. This is a fundamental aspect to be considered: all the ICT solutions within a HIS, such as the EMRs or the EPRs should be able to interact each with the other and with the pre-existing modules already adopted by the single divisions of the organisation. It not reasonable, in fact, to think at the HIS as a homogeneous solution implemented by a single party; on the contrary, it is usually composed of a heterogeneous set of ICT modules implemented by different subjects, adopting different technologies and using different approaches to acquire, manage and exchange data.

This is a widespread problem in the medical context and several initiatives have been undertaken at international level, to standardise roles, protocols, procedures and messages and thus guaranteeing a better integration among modules that are compliant with these standards. Relevant examples are the so called HL7 (Health Level 7) and the IHE (Integrating the Healthcare Enterprise) initiatives.
Main goal if these initiatives is to meet the need of integration, interoperability and sharing of data that become more and more important for medical companies and healthcare structures, due to the heterogeneity of the existing informative systems.
In particular, the goal of the Integrating the Healthcare Enterprise (IHE) initiative is to stimulate integration of healthcare information resources to improve clinical care. IHE develops and publishes detailed frameworks for implementing established data standards to meet specific healthcare needs and supports testing, demonstration and educational activities to promote the deployment of these frameworks by vendors and users.
To pursue these activities, IHE engages the efforts of numerous stakeholders, including care providers, medical and IT professionals, professional associations and vendors. As the initiative has continued to expand worldwide and to encompass a growing number of clinical domains, it has becomes increasingly important to agree upon a general framework for the participation of these stakeholders that is sufficiently well defined to provide effective communication and cooperation.
On the other side, Health Level 7 is a standards-setting organization accredited by the American National Standards Institute (ANSI). They have developed communication protocols widely used in the United States, with growing international recognition and implementations. Its mission is to provide standards for the exchange, management, and integration of data that support clinical patient care and the management, delivery, and evaluation of health care services. This encompasses the complete life cycle of a standards specification-development, adoption, market recognition, utilization, and adherence. The HL7 specifications are unified by shared reference models of the health care and technical domains.

The HL7 version 2.4 messaging standard is currently in use, and version 3, which represents several fundamental changes to the HL7 messaging approach, is in an advanced stage of development. Many people know of HL7 as an organization that creates health care messaging standards, however HL7 is also developing standards for the representation of clinical documents (such as discharge summaries and progress notes).

In this context, Synapsis studied and developed its own vision of an integrated and interoperable HIS. Figure 10 shows the Synapsis approach within which the EMR that will be adopted in HF has been developed.

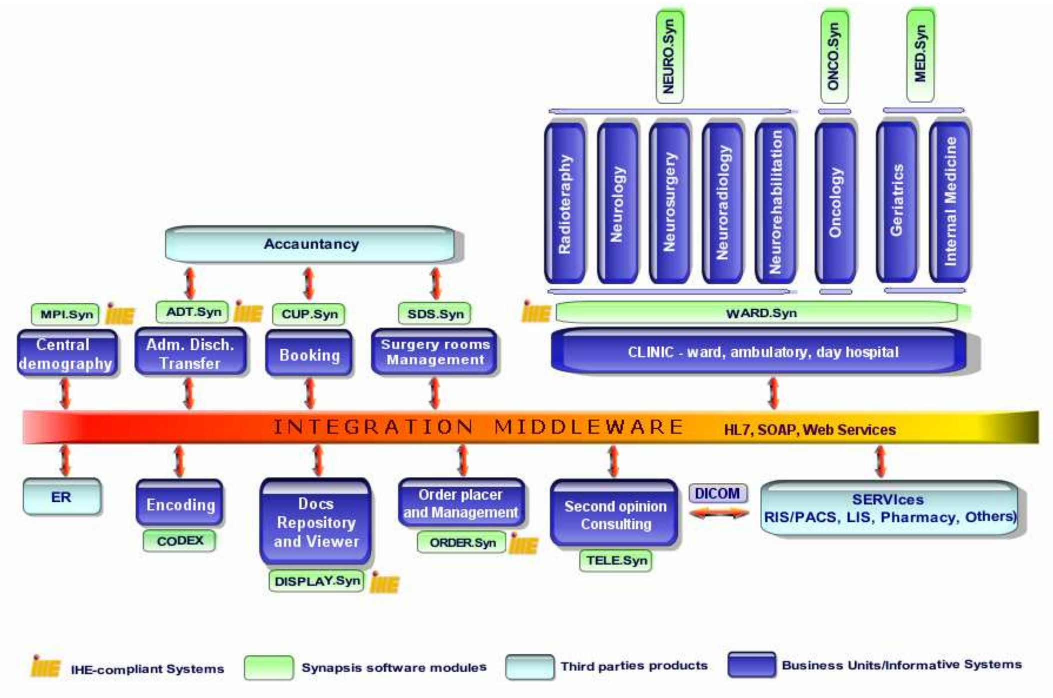


Figure 10 - Synapsis approach for the EMR that will be adopted in the HFP

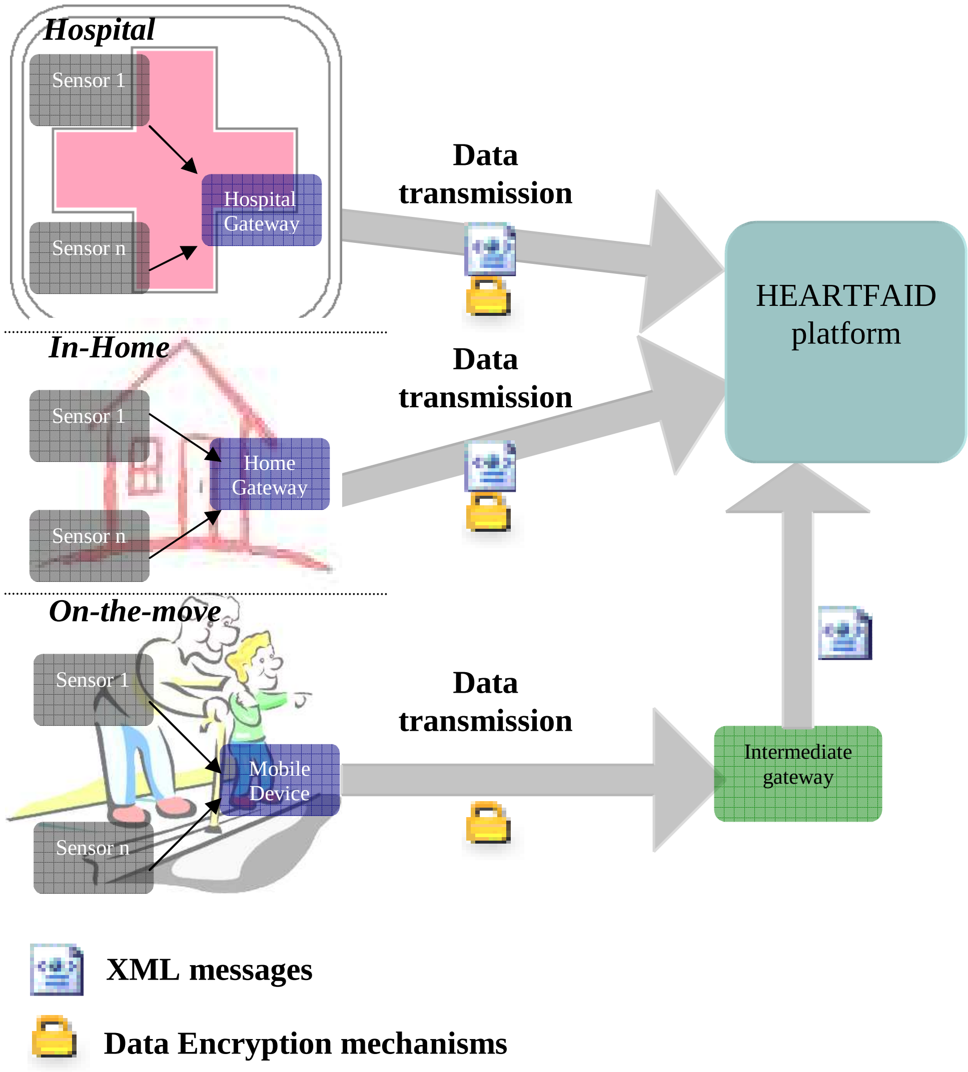
The HF Middleware will have to take into consideration all these aspects in order to guarantee interoperability among the several components that will be integrated into the platform of services.

This entire architecture has a further important requirement to be considered: there must be a common Master Patient Index (MPI), which can be either centralised or most probably distributed, fully integrated with the other modules and accessible by all the EMR/EPR and the EHR as well. This is a mandatory requisite because on this requirement depends the ability to univocally identify a patient involved in the HF monitoring in all the other modules (EPR or EMR) that have information on that patient; in other words, having a patient A, it is necessary that all the clinical information available on this patient and spread among different modules or even among different healthcare structures can be recovered and, overall, that they are related without any doubt to the patient A and not, for example, to any homonym patient.

7 Specifications of the system's components

7.1 The overall architecture of the data acquisition and transmission infrastructure

The overall architecture of the data acquisition and transmission infrastructure is depicted in the following figure.



On the left part of the previous figure we can see the three healthcare environments, namely the hospital environment, the In-home environment and the on-the-move environment. In each of these environments we consider a set of sensors which connected to the respective gateway in order to transmit the measurements. The gateway is responsible to collect the biomedical data from the sensors and transmit it to the HEARTFAID platform over a network connection. Before their transmission, the data is encrypted and transformed into XML messages called observations. In the following sections a detailed description of the system is given.

7.2 Sensor equipment

The sensors equipment which will be integrated into the HEARTFAID platform imposes the examination of many issues related to connectivity, communication protocols, data formats etc. Generally, according to each healthcare environment the sensors which are used can be classified regarding their connectivity as follows:

Healthcare environment	Wireless Connectivity	Wired Connectivity
Hospital	√	√
In-Home	√	√
On-the-move	√	X

7.2.1 Wireless connectivity

The sensors should be available to communicate using wireless channel in an in Home environment, outdoors environment (mainly). As it was seen in previous sections, Bluetooth Technology covers most of the requirements for wireless connectivity, power consumption, and moreover it allows easy setup of a wireless sensor network offering efficient device and service discovery mechanisms. For this reason, it is envisaged that most of the Wireless communication needs in HEARTFAID will be covered via the use of the Bluetooth Technology.

7.2.2 Wired connectivity

Sensors which can communicate their data over a serial wired connection can be used in the Clinical and in-Home environments. The sensor is connected via a serial cable to the data collection device (usually a PC) and uploads its data.

7.2.3 Communication protocols and data formats

A crucial parameter which is taken into account when we consider the communication with sensors is whether the communication protocol is known in order to develop a low-level communication mechanism with the given sensor. In many cases, the sensor vendors do not give access to the communication protocols. In those cases, the reading of the measurements is performed by vendor specific software and the only way to import those readings to the HEARTFAID platform is via parsing output file generated by vendor specific software and transforming them to valid XML observations to be transmitted to the HEARTFAID platform.

7.3 Data Collection devices

7.3.1 Operating System

Regarding the data collection devices (denoted as gateways in the figure), the first parameter which has to be defined is the operating system. In each case, the operating system has to be able to support programmability in order to allow for development of low-level communication mechanisms with the sensors.

Healthcare environment	Operating System
Hospital	WinXP
In-Home	WinXP
On-the-move	Symbian

In the clinical and in-home environments, a PC running WinXP can satisfy the requirements for programmability support. Regarding the on-the-move environment, devices supporting Symbian OS were selected (Nokia 6630 and Nokia N80 smart phones).

7.3.2 Programming Environments

Healthcare environment	Programming Environment
Clinical	Java
In-Home	Java
On-the-move	J2ME

7.3.3 Supported APIs

a) For connectivity with sensors

Healthcare environment	API
Clinical	JSR82, Java Serial IO APIs
In-Home	JSR82, Java Serial IO APIs
On-the-move	JSR82

b) For XML creation and parsing

Healthcare environment	API
Clinical	JAXP, Xerces2, JDOM etc
In-Home	JAXP, Xerces2, JDOM etc
On-the-move	kXML

c) For data encryption

Healthcare environment	API
Clinical	Java Encryption API, Bouncy Castle API etc
In-Home	Java Encryption API, Bouncy Castle API etc
On-the-move	Bouncy Castle API

d) For communication with the platform over IP

In all the healthcare environments, the communication can be carried out via the standard networking functionality provided by Java

7.4 Specifications of the intermediate gateway

Generally, regarding the on-the-move environment, although the creation of XML messages can be performed at the mobile phone's side, the transmission of XML messages usually involves transmission of larger volumes of data. For this reason, for the case of the on-the-move environment, the data is sent to an intermediate server, which is responsible to create the XML messages from these data and transmit them to the HEARTFAID platform. The service logic is implemented using ASP.NET service components and the XML messages are created using the standard API provided by the Microsoft.NET framework.

7.5 Specifications of the HEARTFAID Electronic Health Record

In order to guarantee that the medical history of a patient can be correctly reconstructed, especially after his enrolment into the HF platform, and with the main goal to define and implement a suitable Electronic Health Record for the HEARTFAID main objectives, it will be necessary that both the EMR and the EPR will be compliant with a set of IHE Integration Profiles that have been accurately examined and selected to this purpose. Otherwise, we would obtain a solution specifically customized for the HF scenario and, overall, with few capabilities to be integrated with other environments or contexts.
To this aim, we have identified four main profiles to which the HF modules should be adherent:

- Enterprise User Authentication: it defines a means to establish one name per user that can then be used on all of the devices and software that participate in this integration profile. It greatly facilitates centralized user authentication management and provides users with the convenience and speed of a single sign-on. User authentication is a necessary step for most application and data access operations and it is a workflow improvement for the users.

- Patient Administration Management: this profile establishes the continuity and integrity of patient data, and additional information such as related persons (primary caregiver, guarantor, …). It coordinates the exchange of patient registration and update information among systems that need to be able to provide current information regarding a patient's encounter status and location. This profile supports ambulatory and acute care use cases including patient identity feed, admission and discharge, and transfer and encounter management, as well as explicit and precise error reporting and application acknowledgment.
- Cross-Enterprise Clinical Documents Sharing: this profile facilitates the registration, distribution and access across health enterprises of patient electronic health records. Cross-Enterprise Document Sharing (XDS) is focused on providing a standards-based specification for managing the sharing of documents between any healthcare enterprise, ranging from a private physician office to a clinic to an acute care in-patient facility.
- Cross-Enterprise Sharing of Medical Summaries: this is a mechanism to automate the sharing process between care providers of Medical Summaries, a class of clinical documents that contain the most relevant portions of information about the patient intended for a specific provider or a broad range of potential providers in different settings. Medical Summaries are commonly created and consumed by electronic medical record systems at points in time of transfers of care such as referrals or discharge.

8 Conclusions

In the framework of the work-package WP2 “BIOMEDICAL DATA IDENTIFICATION AND COLLECTION” of the HEARTFAID project, this document has dealt with the analysis of the requirements and specifications of the Data Acquisition and Transmission Infrastructure, which will be responsible to gather the biomedical data in numerous healthcare environments and transmit them to the HEARTFAID platform.

Initially, we analysed the relevant technological background aiming to identify the enabling technologies that are involved in the design and development of the Data Acquisition and Transmission Infrastructure. Since the current practice in the field revolves around the concept of monitoring of patients in their own environment, the technological analysis mainly focused on wireless sensor networks and ireless technologies in general, however it was stressed that compatibility and coexistence with the already installed systems (e.g. devices in hospital using, for example RS-232 interfaces) is a highly desirable feature. Furthermore, the technological analysis included current practices related to the storage of data and identified some critical points related to the current practices on this field.

After having performed the technology analysis on the field, the requirement of the system were defined, starting from the identification of some general requirements and proceeding with the presentation of some more specific requirements related to the identified healthcare environments (clinical, in-home and on-the-move). These requirements were built having in mind the use of existing modules and technologies, the ease of use, the extensibility of the system (i.e. ease in adding new devices in the infrastructure) as ell as the security of data. With these requirements in mind, the set of elements that comprise the infrastructure was defined and detailed requirements were presented for each one of these elements. These elements include medical devices, communication modules and communication protocols, data collection devices, software modules for communication with the devices, data processing and data transmission, as well as the EPR for the storage of data. In the scope of the detailed presentation of requirements a more detailed presentation of some key technologies was performed, aiming mainly to demonstrate how the characteristics of these technologies can match the design goals of the Data Acquisition and Transmission Infrastructure. For example, Bluetooth is the technology of choice in implementing wireless communication with medical devices, Java and J2ME with XML and cryptography support is the technology of choice for data processing enhancement and security, etc.

Finally, having defined the requirement of the building elements of the infrastructure, an overall architecture of the infrastructure was presented and major specifications for its modules ere given (more detailed specifications will be given in D19 – Prototype of the Data Acquisition and Transmission Infrastructure). More specifically, the infrastructure will be able to operate in the three identified healthcare environments, and will adopt an approach according to which the set of devices in each environment will communicate with a localised gateway for the clinical and home environments, while in the on-the-move environment the sensors will communicate with a smart phone in order to upload their data. The data are then processed and enhanced utilising available APIs in the environment of the gateways (e.g XML APIs for composition of structured messages, cryptography API for the encryption of data etc) and is then transmitted over IP to the HEARTFAID platform utilising public fixed or wireless networking infrastructure. Especially in the case of the on-the-move environment, and in order to minimise the volume of the transmitted data, the collected data are sent over IP to an intermediate gateway, where the data enhancement process is performed and the final message is sent to the HEARTFAID platform.

9 Bibliography and References

[1] Emil Jovanov, "Wireless Technology and System Integration in Body Area Networks for m Health Applications," Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Shanghai, China, September 2005.

[2] Nikolay Dokovsky, Aart van Halteren, Ing Widya, "BANip: Enabling Remote Healthcare Monitoring with Body Area Networks", FIDJI 2003: 62-72

[3] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, A. Wood, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges", Workshop on High Confidence Medical Devices Software and Systems (HCMDSS), 2005

[4] Richard Bults, Katarzyna Wac, Aart van Halteren, Victor Nicola, Dimitri Konstantas, *Goodput* "Analysis of 3G wireless networks supporting m-health services", proceedings of ConTEL 2005 - 8th International Conference on Telecommunications, 15-17 June 2005.

[5] J. B. and C. F. Sturman, "Bluetooth 1.1: connect without cables," Pearson Education, 2001

[6] W. H. Inmon, "Building the Data Warehouse,"3rd ed., John Wiley Publishing, 2002.

[7] Carl Falcon. Wireless medical devices: Satisfying radio requirements. Medical Device & Diagnostic Industry, page 80, September 2004. URL: http://www.devicelink.com/mddi/archive/04/09/018.html

[8] National Association for Home Care. Basic statistics about home care 2001. URL: http://www.nahc.org/Consumer/hcstats.html

[9] Population Projections Branch U.S. Census Bureau, Population Division. National population projections, population pyramids. URL: http://www.census.gov/population/www/projections/ natchart.html, Oct.12, 2004.

[10] Poropudas T. Medical handheld usage to triple by 2005. Helsinki, April 02, 2002 05:57 GMT, Mobile CommerceNet, http://www.mobile.commerce.net.

[11] Poropudas T. Arbonaut develops medical location with VMWS. Helsinki September 18, 2001 06:08 GMT, Mobile CommerceNet, http://www.mobile.commerce.net

[12] Istepanian RSH, Tachakra S, Banitsas KA. Medical Wireless LAN Systems (Med-LAN). State of the Art, Challenges, and Future Directions, Proceedings of the 3rd International Conference in the Delivery of Care, E-Health; a Future Prospective, City University, London, April 2001:43–49.

[13] Istepanian RSH, Chandran S. Enhanced telemedicine applications with next generation of wireless systems. Proc. First Joint BMES/EMBS IEEE International Conference of Engineering in Medicine and Biology, Atlanta, Oct 1999:708.

[14] Pattichis CS, Kyriacou E, Voskarides S, Pattichis MS, Istepanian R, Schizas CN. Wireless Telemedicine Systems: An Overview. IEEE Antenna and Propagation Magazine 2002;44:143–153.

[15] McDonough D Jr. Wireless special report: Healthcare goes wireless. Wireless News Factor, April 9, 2002.

[16] Kenneth P. Fishkin, Kurt Partridge and Saurav Chatterjee, “Wireless User Interface Components for Personal Area Networks”, IEEE Pervasive Computing, vol. 1, Oct-Dec 2002, pp. 49-55.

[17] Konrad Lorincz, David J. Malan, Thaddeus R.F. Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoffrey Mainland, Matt Welsh and Steve Moulton, “Sensor Networks for Emergency Response: Challenges and Opportunities”, IEEE Pervasive Computing, Special Issue on Pervasive Computing for First Response, Oct-Dec 2004, pp. 16-23.

[18] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J.A. Stankovic, “An Assisted Living Oriented Information System Based on a Residential Wireless Sensor Network”, Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference Arlington, Virginia, USA, April 2-4, 2006

[19] Pyper C., Amery J., Watson M. and Crook C. (2004) Medical Science Monitor, 10, Sr17-Sr22.

[20] ESA (2004) Telemedicine 2010: Visions for a personal Medical Network, The European Space Agency, July 2004

[21] Eysenbach, G. (2000). British Medical Journal, 320, 1713-1716.