HEARTFAID

D11 – Functional specifications of the Middleware

Submission date:
Due date of document: 01/12/2006

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 - 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)

D11 – Functional specifications of the Middleware

Document summary	
Document title:	D11 – Functional specifications of the Middleware
Document classification:	Derivable D11
Dissemination level:	CO
Submission date:	22 January 2007
Due date:	01 December 2006
Authors:	Christos Biniaris – VMW Stelios Louloudakis – FORTHNET Sergio Di Bona – SYNAP Massimo Martinelli – CNR Franco Chiarugi, Vangelis Sakkalis – FORTH
Work package:	WP3 – Middleware, Interoperability and Integration
Report version:	1.7

Short description
This deliverable has the main goal to analyse the problems that should be faced to achieve the Grand Vision of the HEARTFAID project, and to define the functional specifications of different components of the Interoperability and Integration Middleware.

Change record		
Version number	Changes	Release date
0.1	Template	29/09/2006
0.3	Preliminary draft	04/12/2006
0.7	First draft	18/12/2006
1.0	Advanced Draft	22/12/2006
1.5	Final Draft	19/01/2007
1.7	Final Version	22/01/2007

Table of contents

Executive Summary

The HEARTFAID (HF) project aims to devise, design, develop and deploy advanced and innovative computerized systems and services that, by collecting, integrating and processing all relevant biomedical data and information, are able to improve medical knowledge and make more effective and efficient all the processes related to diagnosis, prognosis, treatment and personalization of the Heart Failure care in elderly patients.

This general goal will be achieved by:

- developing an innovative technological platform for informative and decision support, which can make the procedures of diagnosis, prognosis and therapy more effective and reliable for the patient and optimal in the use of medical and clinical resources. This platform, by exploiting innovative results on computational modelling, knowledge discovery methodologies, visualization and imaging techniques, and using the medical knowledge of the relevant domain, is able to effectively integrate and process biomedical data and information at different levels of structure;
- defining new health care delivery organization and management models for the relevant domain, which may result in more effective and efficient use of the needed total resources (health care operators, health care equipments, financial resources).

A key work-package to achieve the first point is WP3 - MIDDLEWARE, INTEROPERABILITY AND INTEGRATION, which has the main goal to analyse, design and implement the core software architecture of the HEARTFAID platform, i.e. the Middleware.

The Middleware is responsible to guarantee the integration and the interoperability among all the modules of the platform, as well as of the services provided to the end-users. As described in the Description of Work (DoW) of the project concerning the architectural design of the platform, the Middleware has been decomposed into two different, but integrated, levels: the Interoperability Middleware and the Integration Middleware. The first level has the main objective to guarantee the interoperability of the HEARTFAID system components and, using the approach of service-oriented integration, to integrate the functionalities the platform will provide to the end users, while the second level has the goal to design and develop a Data Management System that is responsible to guarantee the following features:

- ➢ all the data flowing within the entire platform are compliant with the standards identified in work-package WP3;
- ➢ management of the heterogeneous repository in order to allow the treatment of raw data, laboratory data, structured information (Electronic Patient Record (EPR), data entry services, and so on), multimedia and other data (reports, images, ultrasound signals, and so on).

The first fundamental step for the attainment of these objectives, is to analyse the requirements of the final users, and hence of the platform, and to define the functional specifications of the platform itself.

This deliverable has the main goal to analyse the problems that should be faced to achieve the Grand Vision of the project, and to define the functional specifications of the different components of the Interoperability and Integration Middleware.

In more details, this document has been organised as follows.
At first, an introduction will provide a general overview about the Health Information Systems. The main international initiatives to standardise protocols, roles, messaging procedures and data encoding procedures, will be described. Moreover, the vision that will drive the design and development of the HEARTFAID Electronic Health Records will be presented.

Afterwards, we will analyse the state of the art and the methodological foundations that will support the "application integration". The issue of interoperability, including data access, exchange, and integration becomes more and more sophisticate especially for medical companies and healthcare structures, due to the heterogeneity of the informative systems available. Therefore, at this level it is nowadays very important to adopt adequate standards for data encoding and communication. These standards and the most common approaches, when they are considered suitable for the purposes of the HF project, will be adopted when defining the main infrastructure of the HF platform of services, i.e. the middleware.

After a description of the state of the art on the most advanced solution able to support interoperability, we will analyse the requirements that the middleware should satisfy. These requirements are mainly concerned with interoperability issues of the healthcare structures. In fact, although health organizations are important users of the Information Technology, each of them and the system implemented typically use their own protocols and adopts different standards, principally because there is a lack of divulgation of information concerning standards. Furthermore, concerning data retention and exchange, information models in use are not always based on standards, and even when they are, such standards are not interoperable and even not enough spread within the stakeholders. Typically those solutions address the needs for local administration and management of services, that when available are not supported by its own ontology. Thus, health information is sparse, and with severe interoperability problems as people move across borders or even across health systems independently of the place.

In the future, interoperability will enable the seamless integration of any Health heterogeneous systems. Thus, health data will be made available independently of the standards adopted for retrieving and exchanging information. A visionary future seeks a methodology to enhance organization's interoperability, keeping the same organization's technical and operational environment, improving its methods of work and the usability of the installed technology through ontological harmonization of the organization's models in use. Procedures will take complete care regarding ethics and privacy. This will allow secure and fast access to comparable public health data and to patient information located in different places over a wide variety of repositories and medical devices.

After the requirements have been identified, we will analyse and define the functional specifications of the Interoperability and the Integration Middleware, as well as the issues related with the integration of the HF Decision Support System. In particular, the different components of the middleware layers will be described. The challenge of determining the functional specifications of each module lies in identifying the user needs and designing and implementing comprehensive and convenient interfaces, both with the final users and between the modules themselves.

Since the Dow of the HEARTFAID project does not foresee a specific document on the design of the Middleware, we made a step forward with respect to the identification of the functional specifications and in Section 6 we will introduce some models that can be adopted in the subsequent implementation of the platform. Preliminary hardware specifications will also be identifies.

Finally, due to the particular context in which the platform of services will operate and the sensibility of the data that will be handled, the security aspects related to safety, secrecy and privacy of data will examined.

1. Glossary of terms

TERM	DEFINITION
3G	Third Generation
ACO	Authenticated Ciphering Offset
ACT	Array Coherence Tomography
ADSL	Asymmetric Digital Subscriber Line
ADT	Admission Discharge Transfer
AECG	Electrocardiographic Ambulatory Holter Monitoring
AmI	Ambient Intelligence
AMR	Automated Medical Record
ANSI	American National Standards Institute
API	Application Program Interface
B2B	Business-to-Business
BPIOAI	Business Process Integration Oriented Application Integration
CDA	Clinical Document Architecture
CEN	European Committee for Standardization
CHF	Chronic Health Failure
CIF	Ciphering Offset Number
CLI	Call Level Interface
CORBA	Common Object Request Broker Architecture
CPR	Computerised Medical Record
CQ	Continuous Queries
DB	Database
DCOM	Distributed ComponentObject Model
DICOM	Digital Imaging and Communication in Medicine
DM	Data Mining
DMS	Data Management System
DoW	Description of Work
DSS	Decision Support System
DTD	Document Type Definition
ECG	Electrocardiogram
EDF	European Data Format
EEG	Electroencephalogram
EHR	Electronic Health Record
EMR	Enterprise Electronic Medical Record
EN	European Norm
ENV	European Prestandard
EPR	Electronic Patient Record
ER	Electronic Record
ES	Expert System

ESB	Enterprise Service Bus
FDA	Food and Drug Administration
GP	General Practitioner
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GW	Gateway
HCI	Human-Computer Interaction
HF	HEARTFAID
HFP	HEARTFAID Platform
HIS	Healthcare Information System
HL7	Health Level 7
HL7 aECG	HL7 Annotated ECG Standard
HTTP	Hypertext Transfer Protocol
ICEHR	Electronic Health Record for Integrated Care
ICT	Information and Communications Technology
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IHE	Integrating the Healthcare Enterprise
IOAI	Information Oriented Application Integration
IP	Internet Protocol
ISHNE	International Society for Holter and Non-invasive Electrocardiology
ISO	International Standards Organization
ISSS	Information Society Standardization System
IT	Information Technology
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
JAXB	Java API for XML Binding
JAXR	Java API for XML Registry
JDBC	Java Database Connectivity
JMS	Java Messaging Service
JNDI	Java Naming Directory Interface
JSR	Java Specification Request
JTA	Java Transaction API
JTS	Java Transaction Service
JVM	Java Virtual Machine
KDD	Knowledge Discovery in Databases
LM	Link Manager
MIME	Multipurpose Internet Mail Extension
MIT	Massachusetts Institute of Technology
MML	Medical Markup Language
MOM	Message-Oriented Middleware
MPI	Master Patient Index
MRI	Magnetic Resonance Imaging
NDA	Non Disclosure Agreement

OBS	Observation
ODBC	Open Database Connectivity
OMF	Observation Message Format
OMG	Object Management Group
ORB	Object Request Broker
OS	Operating System
OSI	Open System Interconnection
PC	Personal Computer
PDA	Personal Device Assistant
PET	Privacy Enhancing Technique
PH	Protocol Handler
PIM	Personal Information Management
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POAI	Portal Oriented Application Integration
QoS	Quality of Service
RDA	Remote Data Acquisition
RID	Retrieve Information for Display
RIM	Reference Information Model
RMS	Resource Management System
RPC	Remote Procedure Call
SCP-ECG	Standard Communication Protocol for Computer Assisted Electrocardiography
SMS	Short Messaging System
SOA	Service Oriented Architecture
SOAI	Service Oriented Application Integration
SOAP	Simple Object Access Protocol
SPECT	Single Photon Emission Computed Tomography
TC	Technical Committee
TCP/IP	Transmission Control Protocol – Internet Protocol
UDDI	Universal Description Definition and Integration
UI	User Interface
VCG	Vectorcardiogram
WDSL	Web Service Description Language
XDS	Cross-Enterprise Document Sharing
XML	Extended Markup Language
XML-RPC	XML-based Remote Procedure Call
XSD	XML Schema Definition

2. Introduction

The medical field, more than other fields, is affected by the proliferation of heterogeneous information systems. The origin of this phenomenon should be searched into the complex organization of the medical structures, usually composed of a high number of departments, operative units, wards and services, which are typically provided with a wide decisional autonomy. So, many separate "islands", not communicating among them, have grown up so far. This has led to an increasing need of developing integrated platforms able to guarantee the seamless interaction among different environments and the interoperability with both traditional and innovative Information and Communications Technology (ICT) solutions, where data can be:

- automatically acquired by pervasive and non-invasive Remote Sensor Networks,
- accessed anywhere and anytime,
- exchanged using standard formats and procedures.

To face these needs, modern approaches include the design and implementation of multilevel distributed platforms that guarantee the interoperability among the components as well as the actors of the system, and provide easily accessible and integrated services. Moreover, these platforms should be able, in the future, to support both decision makers and clinicians in the processes of diagnosis, prognosis, treatment and personalization of healthcare assistance, with the main goal to guarantee a better quality of life to pathological patients and to reduce the number of hospitalisations thus decreasing both the social and the economical impact on the healthcare system.

For the time being, we can state, with good certainty, that the achievement of a so ambitious objective will be possible assuming the following requirements:

1. supplying availability and easy access to heterogeneous patients data;
2. designing common user interfaces for integrated and easy-to-use services for healthcare professionals;
3. supplying availability and easy access to formalised clinical knowledge (declarative knowledge, procedural knowledge, and new discovered knowledge).

The third point has a crucial importance, as stated in a white paper of OpenClinical web portal: "It is now humanly impossible for unaided healthcare professionals to possess all the knowledge needed to deliver medical care with the efficacy and safety made possible by current scientific knowledge. This can only get worse in the post-genomic era. A potential solution to the knowledge crisis is the adoption of rigorous methods and technologies for knowledge management. (...)".

In order to develop an integrated and interoperable system, able to guarantee an umbrella of services that range from the acquisition, sharing and management of raw or structured data to the provision of effective diagnostic support to clinicians, it becomes necessary to implement multi-level heterogeneous and distributed architectures where each level has different responsibilities and provides integrated functionalities to the adjacent levels.

The levels of a general comprehensive architecture can be outlined as follows:

- Data level (Data collection and transmission): this is the lowest level, concerned with the source data. This level is responsible for collecting all data that can be exchanged with the external world, including raw data, structured and laboratories data, non-structured information and multimedia data.
- Middleware level (Interoperability/Integration Middleware and Repository): the data level interacts with the middleware level, which is responsible for the exchange of data among the modules of the system and it is in charge of guaranteeing the interoperability both inside the architecture and outside with the external end-user world. In addition, this level certifies that all the incoming, outgoing and exchanging information, as well as all the communications performed within the system and between the system and external applications are compliant with the standards for clinical data representation and communication.
- Knowledge level (Data preparation, Knowledge Discovery in Database and Ontologies): this level deals with the management of the domain expertise and know-how, both explicit (i.e. formal know-how already represented using a formal approach, e.g. a clinical protocol) and implicit (i.e. derived from the daily practice of the clinicians and their experience), as well as with the extraction of novel, useful and non-trivial knowledge from the system repository by using innovative knowledge discovery processes.
- Decision Support level (Decision Support System and Signals/Images processing): this level is typically based on Knowledge Bases and Ontologies and provides an effective support to the daily practice of the clinicians by implementing adequate data processing algorithms, providing guidelines to medical protocols as well as diagnostic suggestions, generating alarms in case of critical situations, and so on. In other words this level takes care of accessing the knowledge base in order to apply rules and to manage the results that are used to perform predefined actions.
  From an operative point of view, the requests generated at the End-User level are processed by the middleware that is able to activate an adequate process in the DSS level. The DSS will interact with the knowledge base level and applying selected rules on the available data (provided by the middleware itself) will be able to return an answer that will be apportunely formatted and forwarded to the End-User level.
  A final remark concerns the possibility of the End-User level to dynamically require real-time signals and images processing in order to extract new information.

A main aspect of the DSS is that it is not a continuously running process but it is rather activated on demand.

- End-users level: this is the higher level of the platform and interacts with the external users, both human being and software application. This level provides specific services and applications to exploit the functionalities of the developed platform.

In this context, this deliverable deals in particular with the Middleware level, which is the component responsible to guarantee the efficient access and exchange of the available data, as well as the integration among the end-user services.

In this document we will afford the problem of defining the functional specifications and the requirements of the HEARTFAID Platform (HFP) middleware, as well as a preliminary design of the middleware itself. Before getting into the core aspects of the functional specifications of the middleware, it is important for the reader to understand the application domain where the HF solutions will be implemented and experimented. To this goal, the following paragraph will provide an introductive description of a Healthcare Information System (HIS) used in a healthcare system/structure, with particular reference to the real scenarios in modern societies and the common problems of these structures.

The content of this document is organised as follows: Section 3 describes the State of the Art of the existing solutions as well as the methodological foundation of the solutions that are likely to be adopted in the realisation of the HF platform of services; Section 4 analyses the requirements of the middleware according to the HF scenarios defined by the entire consortium; in Section 5 we will describe the functional specifications of both the interoperability and the integration middleware; in Section 6 we will propose a possible design of the middleware that will be implemented; finally, Section 7 will discuss about the security issues that should be faced when implementing the platform.

2.1 The Healthcare Information System (HIS)

The healthcare structures are now 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.

There are many initiatives in this direction and many healthcare centres are now being equipped with LAN/WAN infrastructures, nevertheless there is not a real integration or cooperation among the different organisations so that data, exams, reports, and any other kind of information related to same patient cannot be exchanged from one place to another. On the contrary, there is an increasing need to share and integrate information arriving from different sources through an adequate underlying ICT middleware and modern technologies already allow to collect data from a source and to make them available to other subjects under security and privacy constraints.

However, nowadays, when a patient has a contact with a healthcare structure, he generates a large amount of data that is typically recorded and exchanged on a paper basis. This is due to the fact that the real potentialities of the ICT are not being fully exploited, especially in the healthcare sector, and only recently we are starting to use these technologies to implement innovative services for both the citizens and the clinicians.

According to our experience, we will describe shortly what the main problems of the actual healthcare scenario are:

- It does not exists a centralised patient demographic index; 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.
- It often does not exist a central booking service so that the schedules of each single department (e.g. Radiology, Surgery, Echo, etc.) are not synchronised 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 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 specifications:

- 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.

As reported in the DoW of the HEARTFAID project, one of the main objectives in the framework of WP2- BIOMEDICAL DATA IDENTIFICATION AND COLLECTION, specifically addressed by task T2.2 - Design and development of the data Acquisition and Transmission infrastructure, 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 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 needs of the 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 very important aspect that shall be taken into account when developing the HEARTFAID EHR will be **integration** with existing modules: within a Health Information System (HIS) of a healthcare structure it is important that all the ICT solutions, such as the EMRs or the EPRs, are 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.

To this aim, 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. The following Figure 2.1 shows the Synapsis approach within which the EMR that will be adopted in HF has been developed.

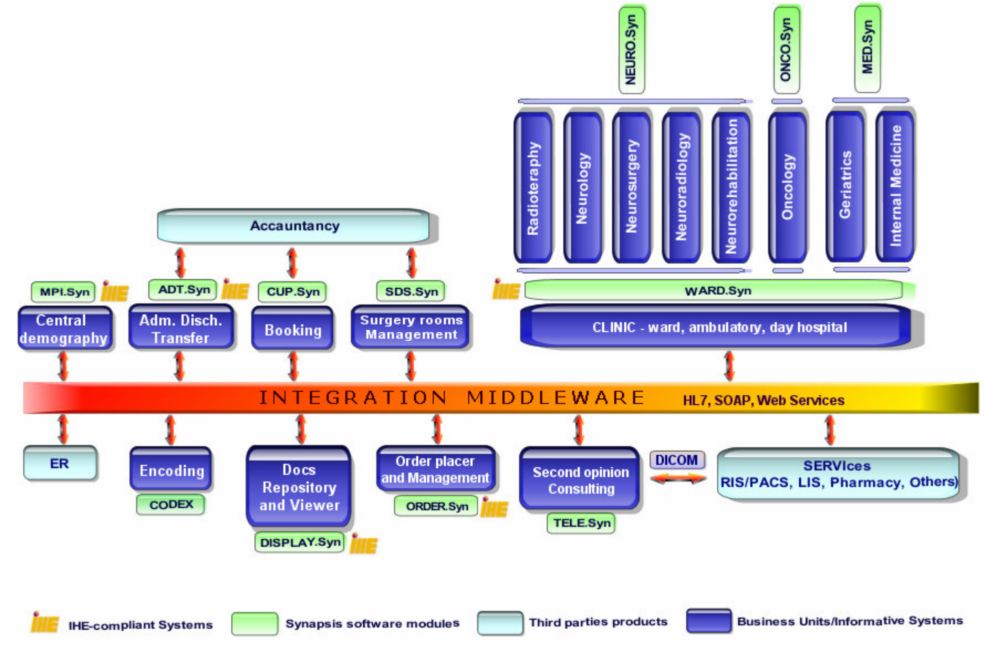

Figure 2.1 - Synapsis approach for the EMR that will be adopted in the HFP

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.

In the following of this document we will analyse the methodological foundations, the requirements and the functional specifications of the Interoperability and the Integration middleware, as defined in the DoW of the HEARTFAID project.

2.2 Bibliography and References

[1] http://www.openclinical.org
[2] IHE Europe Organization: http://www.ihe-europe.org
[3] Health Level Seven (HL7) Standards Developing Organizations: http://www.hl7.org

3. State of the Art and Methodological Foundation

The issue of interoperability, including data access, exchange, and integration becomes more and more sophisticate especially for medical companies and healthcare structures, due to the heterogeneity of the available informative systems. Therefore, at this level it is nowadays very important to adopt adequate standards for data encoding and communication (e.g. the "Health Level 7"-HL7 standards, the "Clinical Document Architecture"-CDA to encode the information, the "Integrating the Healthcare Enterprise"-IHE initiative to standardise the interaction among the modules of the HIS, and so on).

The problem is more complex than simply adopting standard languages to communicate among different applications. Institutions working on these issues are addressing also the problem of defining standard roles and workflows in order to describe accurately typical interactions inside healthcare structures and across different facilities. It is needed a totally interoperating and integrated environment for healthcare structures, able to exchange clinical data, to access distributed and shared repositories of both raw data and complex information, as well as to access computational resources and medical expertise, etc.

Application Integration is defined as the uses of software and computer systems architectural principles to integrate a set of computer applications. There are various approaches to Application Integration, and also the above definition is quite generic, nevertheless we can agree with the following general **categories**:

- Information-Oriented (IOAI)
- Business Process Integration Oriented (BPIOAI)
- Service Oriented (SOAI)
- Portal Oriented (POAI)

Having the "Application Integration" as a main problem to be faced, the following sections will describe the main aspects of this issue.
In particular, a first topic to be afforded is concerned with the topologies that can be implemented to support the application integration.

Afterwards, we will analyse the technologies that can be adopted to realise a suitable platform for Application Integration. Finally, we will describe the specific methodological choices in the HF context.

3.1 Topologies for Application Integration

The main **topologies** that can be utilized in application integration are: Point-to-point, Hub-and-spoke and its variation as Multihub configuration, or Bus-like topology.

In the Point-to-Point application integration topology each application has a communication channel toward another application (see Figure 3.1 A). This kind of integration is the simplest it can be implemented because frees the architect and developer from dealing with the complexity of adapting to common interfaces, typically designed to manage generic applications; but it is the worst solution in terms of scalability and is applicable with a limited number of applications only.

The Hub-and-spoke is the most traditional approach; it consists of a star like topology where applications are on star tips and information are brokered through the hub, a centralised application that performs data mapping, transformation and message routing (see Figure 3.1 B). The main advantages of this centralised solution are:

- a lowest connection complexity, that is *O(n)* (we have N-1 as number of connections),
- a better manageable administration, because of the one only centralised application,
- simpler spokes, that have to manage only one connection.

The drawback of this approach is that the centralised application can become a bottleneck and a single point of failure; moreover it is not easily scalable. These issues are only partially solved by a Multi-hub configuration.

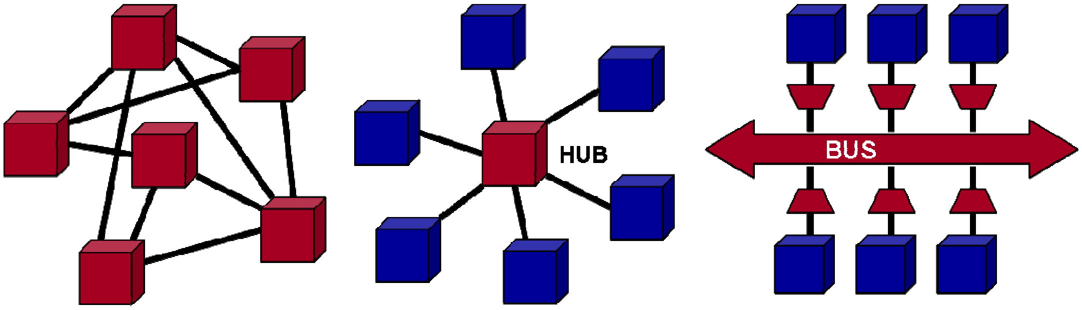

Figure A: Point-to-point **Figure B: Hub-and-spoke** **Figure C: Bus like topology**

Figure 3.1 – Common topologies for Application Integration

The natural topologies evolution is the bus like topology where there is no hub for information exchange and the integration logic is distributed in the endpoints connected to bus (see Figure 3.1 C). The key features of this solution are:

- it improves the theoretical scalability,
- it enables the selective deployment, that is the deployment of the necessary integration functionalities only,
- can route or transform messages conditionally, based on a non-centralized policy,
- the present existence of standard protocols, messages, and technologies, makes the architecture operating-system and programming-language agnostic, besides it reduces proprietary solutions and the vendor lock-in,

- incremental changes can be applied with zero down-time.

For these reasons, in HEARTFAID platform we will adopt a Bus-like topology to integrate applications and systems.

3.2 Middleware

Middleware and standards are technologies that make application integration achievable. By definition middleware is a mechanism that allows one entity (application, database, or system) to communicate with another entity or entities, or in other words it is a software layer that makes easier communication between two or more software systems.

A middleware can have a **point-to-point** or a **many-to-many** configuration. The first one is the simplest, where a pipe links an application to another, and they communicate exchanging a message, or sets of parameter, or files, or data. This model is not suitable in complex contexts, where there are more than few applications that need to communicate. In this case it is difficult to control communications and in general to scale and to maintain the system. In heterogeneous environments (such as in the domain of HEARTFAID) the better configuration is many-to-many where all applications are connected to one centralized and shared server and information are delivered or shared out by way of a broker.

A middleware can provide one or more **communication** models and mechanism:

- Synchronous or Asynchronous, if, respectively, the information sender has to wait or not the receiver response;
- Direct communication, if middleware passes information directly from calling application to remote application; this is also a synchronous communication mechanism;
- Queued communication, if messages produced by an application are put in a queue where the consumer application will take away; typically this is an asynchronous mechanism and does not require the applications are up and running at the same time, on the contrary of the direct communication;
- Publish/Subscribe, if a message published by an application on a topic is delivered by middleware to all applications that are subscribed to the same topic;
- Request/Response, if a middleware allow one-shot communication, where a request is fulfilled by the response of an application server;
- Fire and Forget, if application connected to middleware can fire (send) messages and forget them without taking take care of the destination and the delivery.

There can be several **types** of middleware, and even if it is difficult to classify them, it can be stated that the main categories are in the following.

Remote Procedure Calls (RPCs) are the oldest type of middleware and allow the invocation of a procedure on the network, enabling the distribution of the logic of an application across the network. These types of middleware are simple and the invocation of remote procedures is transparent to the developers' programs, but their proprietary implementations that exists in the market and the low performance because of intrinsic level of processing power required, discourage their use in HEARTFAID.

Object Request Brokers (ORBs) are other well known types of middleware, which enable the objects of an application to be distributed and shared across heterogeneous networks. On the market the main types of distributed objects middleware are: Common Object Request Broker Architecture (CORBA) CORBA and Distributed Component Object Model (DCOM). CORBA is a specification created in 1991 by Object Management Group (OMG) that consists in a set of rules that developers have to follow to become a CORBA-compliant vendor. CORBA is heterogeneous because does not make specific hypothesis on programming languages and platforms and it exists many CORBA-compliant implementation of different producers. DCOM is a standard, created by Microsoft, defining the rules for developers to create DCOM-enabled applications. Even if there are some implementations on non-Microsoft platforms it can be considered homogeneous and working only on Windows platform, therefore inappropriate in HEARTFAID because of the heterogeneous nature of this platform.
On the other hand, even if CORBA promises much, it has a lot of criticism and has failed its spread. The main issues are: the high cost of commercial CORBA implementations, the steep learning curve and the complexity of use; on the other side the diffusion of XML-based exchange of data and web services, has definitely darken CORBA middlewares.

Database-Oriented middleware is the simplest type of middleware; it simplifies communication with a database using Call Level Interfaces (CLIs). The most known CLIs are Microsoft's Open Database Connectivity (ODBC) and Sun's Java Database Connectivity (JDBC).

Message-Oriented Middleware (MOM) is a type of middleware that bases distributed communication on messages exchange. This type of middleware has some useful advantages and will be adopted in HEARTFAID.

Once we understand what is meant with the term "middleware", we immediately figure out that the main features required to the technologies that will be implemented are *interoperability*, of course, and also *portability*. With the requirement of portability, adequate technologies to be exploited are the Java-based solutions and, more in general, the open-source approaches.

Java programming language and Open Source

Java is a programming environment based on an object oriented language that is designed to be portable across the vast majority of computing platforms, from servers to palmtops. Java Virtual Machines (JVMs) that store the runtime support for applications binaries are employed to guarantee this kind of portability.
Java has a native support for network programming that makes it particularly suitable for writing large, distributed applications, which involve many heterogeneous platforms and software agents.
The implementation of HEARTFAID middleware will adheres completely to the Java development framework, both for the code of the software written by the developers and for the external libraries used.
Another important design and implementation choice is the full reliability on open source software. From a conceptual viewpoint, this choice makes it possible for the developing team to rapidly develop and test solutions that use external software tools for visualization, communication or serialization and storage in databases. Besides the intrinsic advantage related to open source solutions like cheapness, big supporting communities for common tools, frequent updates and releases, another major advantage is that the software, support and documentation is quickly available by simply searching the world wide web for it.

3.3 Methodological Foundation in the HEARTFAID Context

As shown in Figure 3.2, the Middleware level of the HF approach is divided into two main layers, following also the most popular approaches of the literature: the Integration Middleware and the Interoperability Middleware. These two layers represent two different abstraction levels of the functionalities of the Middleware.
The Integration layer uses an information-oriented integration approach to communicate and exchange data with all sources. The external Middleware layer, the Interoperability Middleware, should guarantee the interoperability of the system components and should integrate the functionalities provided to the end users.

The methodological choices related to each layer will be discussed in the following two sub-sections.

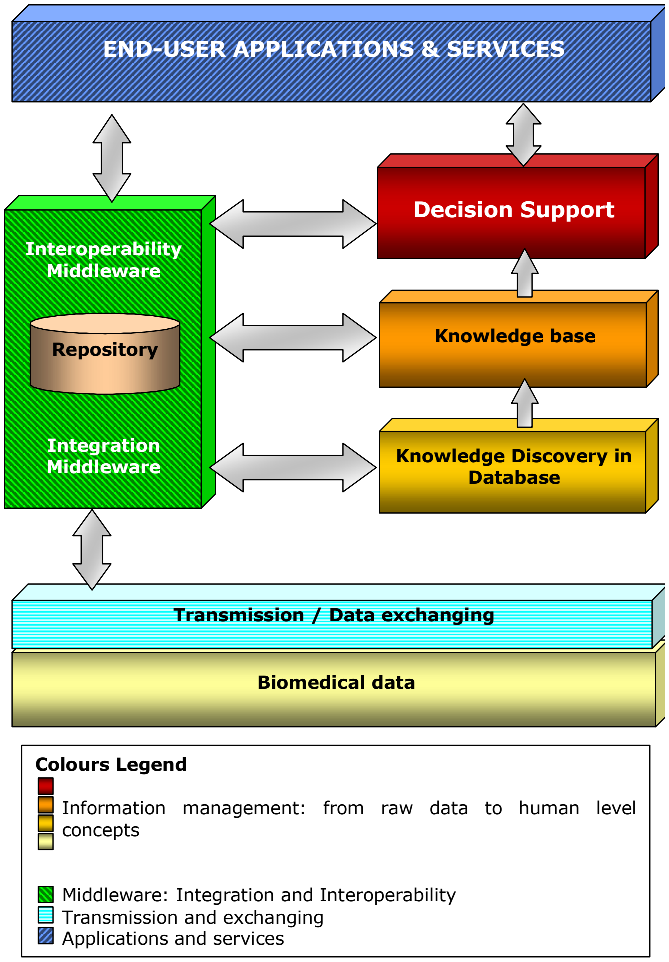
END-USER APPLICATIONS & SERVICES
Interoperability Middleware
Repository
Integration Middleware
Decision Support
Knowledge base
Knowledge Discovery in Database
Transmission / Data exchanging
Biomedical data
Colours Legend
🟥 🟧 🟨 ▨ Information management: from raw data to human level concepts
🟩 Middleware: Integration and Interoperability
▨ Transmission and exchanging
▨ Applications and services

Figure 3.2 – Architecture of a multilevel interoperable platform

3.3.1 Integration Middleware layer

HEARTFAID platform components will be connected through a particular communication infrastructure which has a Bus-like topology.
This topology will satisfy some key requirements for HEARTFAID architecture particularly on abstraction, automatic composition, scalability and evolution. In other words the solution of the bus stems toward a horizontal integration and extensive cross layer interactions that break open the interfaces between layers.

The bus implementation will be set up by a **Message-Oriented Middleware (MOM)** which enables distributed communication that is loosely coupled, reliable, and asynchronous. A MOM specifies two messaging models: Point to Point (based on queues) and Publish-Subscribe (based on topics). Typical implementations of MOMs guarantee messages to arrive in order inside a serial session but no specification is given for message integrity and confidentiality which are left to the application or to specific providers.

In HEARTFAID the MOM middleware will rely on a Java Messaging Service (JMS)-compliant implementation of the Sun JMS specification.
JMS defines set of classes and interfaces and associated semantics that describe how a software agent connects to messaging services. JMS follows the specification / provider pattern where a small subset of entities and functionalities is specified to be restrictive whilst another part is left to the implementation specific choices.
JMS specifies two messaging models: queued (based on queues) and publish/subscribe (based on topics). For example, JMS specifies an addressing scheme based on destination names leaving the possibility to implement more concrete addressing schemes to the provider. Moreover messages are guaranteed to arrive in order inside a serial session but no specification is given for message integrity and confidentiality which are left to the application or to specific providers.
As a last remark, the JMS specification relies on other Java specification for implementing different features. It uses JDBC for implementing message persistence based on common database software. JTA and JTS (Java Transaction API and Services) are used for transactional operations. It uses a component model based on Enterprise Java Beans. Finally JNDI (Java Naming Directory Interface) is used for naming destinations.

The bus like topology can be seen at higher level, and precisely at service level. In fact, **Enterprise Service Bus (ESB)** technology provides a highly distributed approach to integration, with unique capabilities that allow individual departments or business units to build out their integration projects in incremental, digestible chunks, maintaining their own local control and autonomy, while still being able to connect together each integration project into a larger, more global integration fabric, or grid.

3.3.2 Interoperability Middleware layer

In order to guarantee the interoperability among the services offered by the HF platform, we will rely on the so called Service-oriented architectures (SOA).
SOA is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications. The individual functions of an application in the HF platform will be modularized and presented as services. What's key to these services is their loosely coupled nature; i.e., the service interface is independent of the implementation. The applications of the platform can be built by composing one or more services without knowing the services' underlying implementations. For example, a service that is provided by the HEARTFAID platform can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.
Service-oriented architectures have the following key characteristics:

- SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WDSL) is the standard used to describe the services.
- SOA services communicate with messages formally defined via XML Schema Definition (also called XSD). Communication among services typically happens in heterogeneous environments (e.g patient's home and hospital environment), with little or no knowledge about the provider. Messages between services can be viewed as a simple format, in order to be easily understandable by the end user (doctors or patients).
- SOA services are maintained in the platform by a registry that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration (UDDI) is the standard used for service registry.
- Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services. In this way, we can define various access levels, according to user's attributes (e.g. doctors and patients).

3.3.2.1 SOA infrastructure

To run and manage SOA applications, we need an SOA infrastructure that is part of the SOA platform. An SOA infrastructure must support all the relevant standards and required runtime containers. A typical SOA infrastructure looks like the diagram presented below:

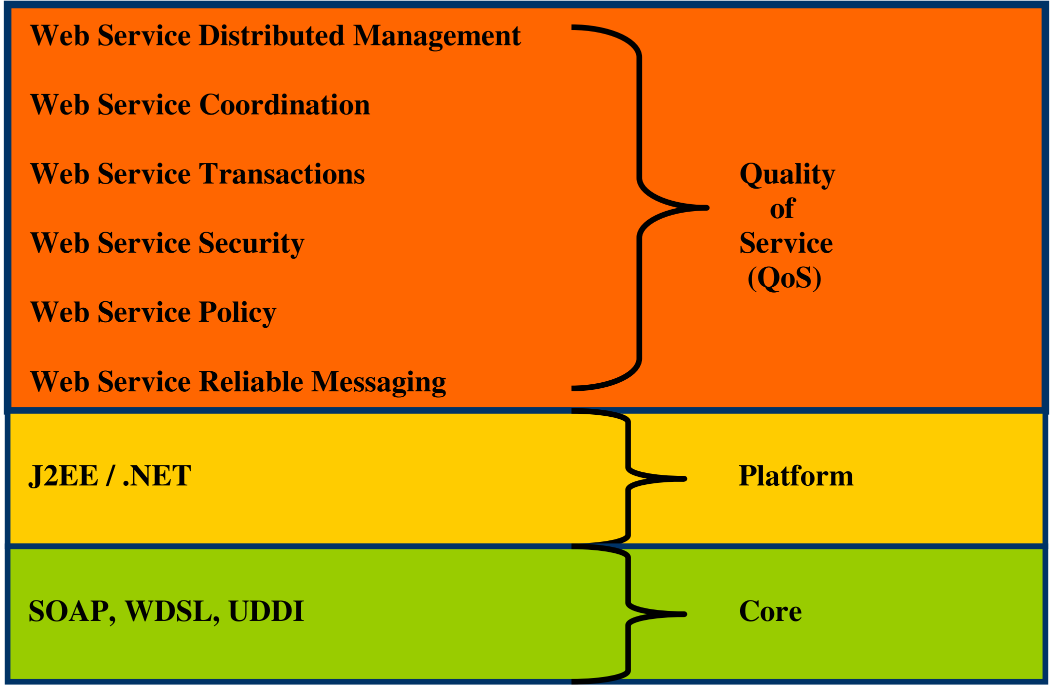

Figure 3.3 - A typical SOA infrastructure

3.3.2.2 J2EE and .Net

Although the J2EE and .Net platforms are the dominant development platforms for SOA applications, SOA is not by any means limited to these platforms. Platforms such as J2EE not only provide the framework for developers to naturally participate in the SOA, but also, by their inherent nature, bring a mature and proven infrastructure for scalability, reliability, availability, and performance to the SOA world. Newer specifications such as Java API for XML Binding (JAXB), used for mapping XML documents to Java classes, Java API for XML Registry (JAXR), used for interacting with the UDDI registries in a standard manner, and Java API for XML-based Remote Procedure Call (XML-RPC), used for invoking remote services, facilitate the development and deployment of Web services that are portable across standard J2EE containers, while simultaneously interoperating with services across other platforms such as .Net.

Java Web Services can be used to develop state-of-the-art web services to implement SOA. The J2EE 1.4 platform will enable us to build and deploy end user web services on the HEARTFAID application platform. It provides the tools, in order to quickly build, test, and deploy web services and clients that interoperate with other web services and clients running on Java-based or non-Java-based platforms.

While the SOA concept is fundamentally not new, SOA differs from existing distributed technologies in that most vendors accept it and have an application or platform suite that enables SOA. It also enables changes to applications while keeping the users of the HEARTFAID platform isolated from changes that happen in the service implementation. SOA enables upgrading individual services of the platform; it is not necessary to completely rewrite an application or keep an existing system that no longer addresses the new HEARTFAID applications requirements. Finally, SOA provides better flexibility in building applications in an agile manner by leveraging existing application infrastructure to compose new services.

3.4 Bibliography and References

[1] http://www.wikipedia.org
[2] http://www.omg.org
[3] D. S. Linthicum. Next Generation Application Integration. Addison Welsey, (2003).
[4] http://cnx.org
[5] http://www.javaworld.com/

4. Requirements of the Middleware

4.1 Introduction and Overview

Health organizations are important users of Information Technology. Nevertheless, typically each Health organization and system uses its own protocols and adopts different standards, principally because there is a lack of divulgation of information concerning standards. Furthermore, concerning data retention and exchange, information models in use are not always based on standards, and even when they are, such standards are not interoperable and even not enough spread within the stakeholders. Typically those solutions address the needs for local administration and management of services, that when available are not supported by its own ontology. Thus, health information is sparse, and with severe interoperability problems as people move across borders or even across health systems independently of the place.

In the future, interoperability will enable the seamless integration of any health heterogeneous systems. Thus, health data will be made available independently of the standards adopted for retrieving and exchanging information. A visionary future seeks a methodology to enhance organization's interoperability, keeping the same organization's technical and operational environment, improving its methods of work and the usability of the installed technology through ontological harmonization of the organization's models in use. Procedures will take complete care regarding ethics and privacy. This will allow secure and fast access to comparable public health data and to patient information located in different places over a wide variety of repositories and medical devices.

Interoperability is one of the most important characteristic of electronic files and information models in eHealth. To reach it, it is necessary to enlarge the collaboration efforts between different institutions, researchers and developers evolved. Hence, the integration and interoperability of systems should go beyond small implementations at hospitals, clinics and even local health systems. It is necessary to enforce interoperability by disseminating and making known proper open standards. This should be the support for open source software to support eHealth. This will result in integrated protocols developed through a framework for mapping over available suitable standards. It is necessary to classify and merge the technology and concepts from the different sources within the domain of applicability, describing them in a unique harmonized structure of classes, attributes, relationships, knowledge components and definitions. Regarding Semantics and Ontology, our vision conducts to a methodology that combines the many ontological worlds in place, i.e., the instances of domain, developing a harmonized ontology aiming to represent a domain of discourse. Conformance testing methods, interoperability checking procedures and the use of web services for universal open and normalized access of electronic health records and health files, are fundamental for the success of the future challenges identified.

4.2 Interoperability issues in the healthcare context

The content of a medical file is a vital piece of information for the patient care and it will be even more important for a subject for whom the need of accurate diagnosis will be determinant for the assessment of his clinical condition and further treatment. Whether it is generated by devices or clinical specialists, clinical information must be accessible to both clinicians and machines.

At this point the concept of interoperability becomes central, as it is related to the ability to share information with other persons or systems. In general terms, the concept of interoperability means that data is retrieved or stored in standardized formats, meaning that authorized people and systems know the data structures used, can read its contents and the information stored has a well-understood meaning. Systems therefore must be interoperable, allowing multiple users to access those records, and allowing information to be shared between health providers. Information must also be shareable between the software of different vendors.

Another concern is interoperability among medical devices; data produced by different devices have different protocols, data structures and even electronic differences. Old devices communicate with RS-232 protocol and some recent devices can be equipped with an IP and are able to connect directly with Internet or some Intranet. Nevertheless, the challenge in terms of devices, from small portable ones to room devices like MRI, would be certainly to have Plug n' Play capabilities. Although communication is the major problem, hardware issues, such as variations in pin configuration and handshaking, can still cause difficulties.

Far beyond the needs for communication between medical devices and clinical information systems, there is the messaging between them as it is important that devices are able to understand the format and content of messages they communicate with each other, in other words, they must speak the same language, both grammatically and semantically.

After all these prerequisites that are of vital importance for Health interoperability, more general aspects should be taken in account like the foundations of a cross-border framework for Health collaboration and healthcare integration. Across all those concerns, architecture should support different eHealth systems and the different players. With adequate middleware implementation, this architecture should assist health professionals in coping with some major challenges, among those, rising demand of services, crescent population aging in Europe, permanent change in healthcare technologies and clinical practice.

Whichever will be the solution adopted in the near future, it should take in account a continuing growth of networks and users and it should integrate the advances in health knowledge providing that; information, training and clinical activities are supported for experts and final users.

4.2.1 Interoperability in eHealth domain

Looking into the existing medical data, it is possible to have a multitude of types of information. Devices produced mostly paper based outputs, like EEG or ECG and others film based outputs like x-rays. Some of the most recent devices, like SPECT or even ACT were producing electronic information that after some processing were stored in paper support. Interoperability of these records depend much on the skills of the health technicians and clinicians to use and process data from different sources and original raw data are most of the times discarded.

When considering patient registration at hospitals and health services, the use of computers is widely spread for administrative proposes and the patient demographics and other personal data is often stored in electronic format. However, there is a lack of interoperability among different devices and different manufacturers.

Usually doctors handwrite notes as they talk with patients and paper is still the most used media for GPs to retrieve clinical information. In most cases, when electronic files are used to store some kind of medical information for some processing or evaluation, after the specific task is completed data is discarded or printed on paper to be stored in archives and being retrieved at a second time. Files are used without temporal concerns; most of the time there is no concern on interoperability as their use is limited at that clinical episode.

The latest devices are producing more and more digital information and that is progressively enabling the storage of electronic data. Another issue to consider is the growth of media able to store electronic information with the subsequent costs reduction, thus allowing comparison and cross-examination among available and historical data.

It is also important to notice the definition from ISO of an ICEHR, Electronic Health Record for Integrated Care. Whilst the ICEHR is the target for interoperability of patient health information and optimal patient care, it should be noted that the large majority of EHRs in use at present are not even shareable as they have own properties and most of the times miss the additional characteristics required to comply with the definition of an Integrated Care EHR (ISO-TC215).

Probably the most important lesson learned in early efforts towards interoperability is that standards can be established by professional use, meaning that if a standard is widely spread and in use by many professionals then it should be a good standard. That is the case of DICOM, the most adopted standard for imaging devices.

4.2.2 Interoperability, future challenges - Officially recognized challenges

Several challenges are open in the field of interoperability and are also recognized by the European Community:

- “Ministers supported concerted actions to address particularly the development of standards enabling interoperability of diverse systems and services and to especially explore the possibilities of open source applications for achieving this objective.”
- “Ministers noted that the full exploitation of the benefits of eHealth technologies requires continued commitment to the development and use of a robust, secure and interoperable infrastructure, as well as to wide

availability and use of broadband communications to maximise the efficiency of eHealth systems and applications."

- "Ministers encouraged the continued investment in research and technological development, ensuring steady advancement of European eHealth technology applications that meet European demands for confidentiality, data security and interoperability."

4.2.3 Electronic recording of data

One of the most important advantages of electronic records is information online delivered on time; the future will certainly demonstrate the advantages of using not only electronic records but a whole framework of information services and support for citizens and clinical practice. The initial costs of implementation of such networks will be meaningless compared with the resulting efficiency and quality of service. Operations that are currently hard to handle like research and analysis will strongly be empowered by interoperable records and devices thus resulting more efficiency, better follow up of distributed information and reduction of costs by avoiding loses or repetition of exams. This interoperable environment will ease clinical practice, help research procedures and deliver more efficient support decision tools. As a corollary of all this , it will be unavoidable to have a more efficient health care systems and better health care for citizens.

It is critical to understand that organizing the delivery of Interoperability is a difficult task. For the naive user, it is all about "standards". However the processes necessary and the level at which the various stakeholders need to be involved is quite different (Report from the CEN/ISSS eHealth Standardization Focus Group).

4.2.4 eHealth Services

For establishing an eHealth platform, one of the important concerns is the services to be implemented in order to provide several functionalities and also to allow the interaction between the different stakeholders, ranging from healthcare business to clinicians and citizens.

The application of ICT to health implies that the following items should be considered amongst the priorities for the application of ICT to a pan-European health environment.

- **electronic health cards including**
    - health record architecture;
    - Health Insurance Cards for proof of entitlement but perhaps containing a medical emergency data set and controlling access to data in a patient's country of residence;
    - promoting the use of health cards generally in the healthcare sector.
    - health data messages
    - management of patient identification including:
        - A common approach to patient identifiers;

        - Access control and authentication.
- **online services such as:**
    - teleconsultation (second medical opinion);
    - e-prescription;
    - e-referral;
    - telemonitoring;
    - telecare.

The ability for such health platforms to provide integration of services in a Healthcare framework will assume a decisive role in the efficiency of a cross-border healthcare network. The major requisites should be those that guarantee a better healthcare for citizens, among those:

- High Performance.
- High Availability.
- Fault Tolerance.
- Security.
- Interoperability of Data and Computers.
- Ubiquitous Access.

4.3 The Middleware in the HF context: Needs and Requirements

According to the HEARTFAID scenarios defined by al the partners of the consortium, the HEARTFAID platform should capable of collecting, integrating, and processing relevant biomedical data and information coming from the main settings actually encountered by patients with Chronic Heart Failure (CHF). In particular, three specific setting have been defined as reference study cases, corresponding to three different healthcare pathways of patients with assessed heart failure:

- the *medical environment*: processes of diagnosis, management, prognosis assessment and medical recommendations are performed on data collected both in the office of the family physician/general practitioner and in the specialized cardiology department with cardiologists involved in outpatient and inpatient care, and with the possibility of running a variety of tests, such as blood tests, ECG, X-Rays, ultrasound imaging studies, etc.;
- the *patient environment*: as in the previous case, patients' data are collected both in the office of the family physician and in the specialized cardiology setting, however, biomedical parameters, relevant symptoms and compliance to prescribed pharmacological and non pharmacological regimens will be monitored at patients' home. Serial measurements of selected biological parameters will be collected by the patients themselves of by their relatives;
- the medical and technological *research environment*: patients' data are collected, in the office of the family physician, in the specialized cardiology setting, and in the ultraspecialized research setting. While supporting the standard CHF management and prognosis assessment, the

HFP will assist in collecting biomedical information for research and development purposes.

In all of the three settings, the middleware should guarantee the easy, everywhere, anytime and safe access to data not only to human operators but also to the different components of platform that might need to recover patient's data for processing. At the same time, the middleware should support the process of information recording in all the scenarios where data might be acquired: patient's home, general practitioner ambulatory, cardiovascular dept., and, when this will be possible according to the hardware specifications, directly from the medical devices used to perform the measurements/exams.

Figure 4.1 shows the patient-centric view of the HF platform: the middleware should allow the interoperability of the main classes of services (i.e DSS, Monitoring, KDD and EHR) with the different contexts taken into account by the reference scenarios of the project. In particular, the middleware should provide the adequate support to allow the end-users to exploit the services offered by the platform, in all the contexts shown in the figure.

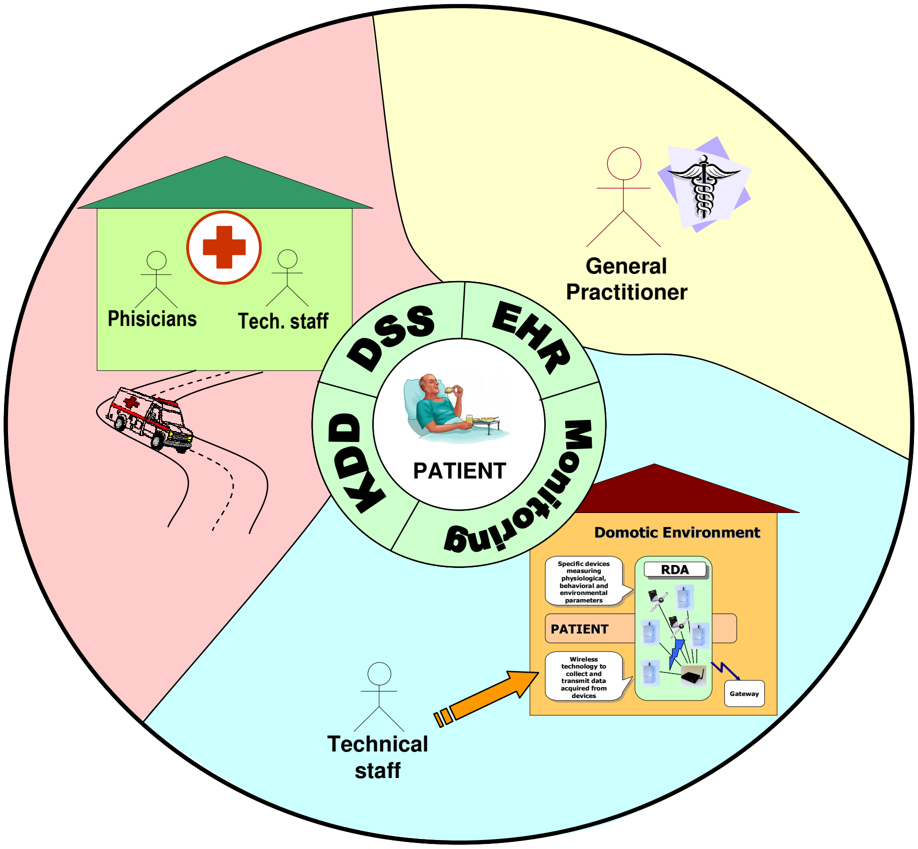
Phisicians
Tech. staff
General Practitioner
DSS
EHR
Monitoring
KDD
PATIENT
Domotic Environment
Specific devices measuring physiological, behavioral and environmental parameters
RDA
PATIENT
Wireless technology to collect and transmit data acquired from devices
Gateway
Technical staff

Figure 4.1 - The patient-centric architecture of the HEARTFAID platform of services

From the figure it is possible to identify what will be the major components that will part of the HF platform or with which the platform will need to be integrated:

- The Health Information System (HIS – See Section 2.1)
- The knowledge management/exploitation system
- The infrastructure that allow the remote monitoring of vital signs and parameters.

The first point, i.e. the integration with the HIS, is very important. In can be resumed as the requirement of interoperability with what can be referred as the "existing external world". The middleware should provide adequate mechanisms to allow the HF platform to interact with existing systems at different levels, such as HIS, EPR, medical devices, and so on.

A first step towards this goal will be the definition of the HEARTFAID EMR, which will later be extended to a more general cardiovascular EPR (see Section 2.1).

At this point we cannot say that the platform will be able to interoperate with any different kind of external system, either complex or simple, since there is a very large amount of different solutions each using different languages, protocols, technologies and architectures. However, we can say that the middleware will be compliant with selected and most common standards for data encoding, transmission and exchange, as well as with standard communication protocols, in such a way as to create the bases for interoperability and to allow interoperability with all the external systems that are compliant with the same standards.

After the implementation of the HEARTFAID EPR, the long term view will be the definition of an Electronic Health Record, according to the definition provided by the Medical Records Institute.

The most important characteristic of the EHR will be the ability to share information between authorised users working in different environments. In technical terms, this requires interoperability of information in the EHR and interoperability of EHR systems which exchange and share this information.

According to ISO, there are two main levels of interoperability of information:

- Functional interoperability – the ability of two or more systems to exchange information (so that it is human readable by the receiver), and
- Semantic interoperability – the ability for information shared by systems to be understood at the level of formally defined domain concepts (so that information is computer processable by the receiving system).

In order to make it possible, several aspects must be taken into account: the overall standards for messaging and storing of information as far as the functional interoperability is concerned, and the ontologies and semantics as far as the semantic interoperability is concerned.

Another important aspect to be considered is the modality for data access. We can distinguish between two main modalities:

- Punctual access: operators or other modules of the platform might need to access specific data pertaining a single patient or even a single exam of a patient;
- General access: KDD processes of statistical services provided by the end-user interfaces might need to access large quantity of data pertaining more patients or even the entire database by applying selection filters.

In both cases the middleware should guarantee security, privacy and availability of data, either data is accessed by a human user or by a software module of the platform such as the DSS or the KDD.

The last point is very important for the goals of the HF project: the remote monitoring of vital signs and parameters. Much of the effort in the design and development of the HF middleware will be focused on this aspect since the ability for the platform to perform continuous monitoring in both the hospital and home care scenarios represent a fundamental added value of this study. The requirements of this component will adhere to the Ambient Intelligence (AmI) principles to which much effort will be spent.

The AmI paradigm stems from the convergence of ubiquitous computing, ubiquitous communication, intelligent user friendly interfaces and distributed low-cost sensing/actuation. AmI is a new paradigm in information technology, in which people are empowered through a digital environment that is aware of their presence and context, and is sensitive, adaptive, and responsive to their needs, habits, gestures and emotions. In the future many e-health applications will improve the quality of healthcare, which will lead to substantial cost savings. For instance, physicians will review radiological films and pathology slides in remote sites, or assist and perform surgery via remote robotics.

First prototypes of AmI systems have been developed, but the realization of true AmI systems requires much additional research of multidisciplinary teams.

4.4 Home Monitoring

For the scenario of home monitoring, the middleware has to support the following features:

a) Collection of biomedical data from the sensors/medical devices.

   In order to enable data acquisition from sensors/medical devices, a part of the middleware has to be implemented on the device which will acquire the data (for example a PDA, a Mobile Phone or a PC). Taking into account that almost every sensor uses a proprietary handshaking and data transmission mechanism, the middleware has to be built in flexible manner in order to easily integrate new sensors into the system.

   According to this approach, the functionality related to low level communication with a sensor (e.g. sending or receiving data over a wired/wireless channel) has to be built into the middleware of the device which performs the data acquisition. Furthermore, this functionality has to be exploited by Java base classes, representing the communication with a sensor in an abstract manner. Consequently, in order to add a new sensor, a Java class has to be created, which inherits the base class and includes all the sensor related communication characteristics.

b) Enhancement of this data with additional information
Apart from the collection of raw data from sensors, the middleware will be responsible to enhance this data with additional information in order to prepare it for transmission to the HEARTFAID platform. For example, it is obvious that a crucial parameter for a measurement is the exact time that the measurement was performed. Many medical devices support time setting, while others don't. However, when the data is gathered, one cannot rely on the proper time settings of a given medical device. For this reason, the middleware should be responsible to create a timestamp for every measurement. Furthermore, the data could be enhanced with information related to the specific user that performs the measurement, alarm conditions for the specific measurement etc.

c) Organisation of data into messages with standard format
In order to allow for adaptability and extensibility of the transmitted data structures, it was agreed to adopt a standard XML based format for the transmission of the data in form of messages, called observations. The middleware should be able to create this standard format transparently from the application.

d) Transmission of these messages to the HEARTFAID platform
Finally, the middleware should be able to handle the communication details with the HEARTFAID platform and report possible communication problems to the end-user application.

The block diagram of the middleware for the Home monitoring scenario is depicted in the following figure:

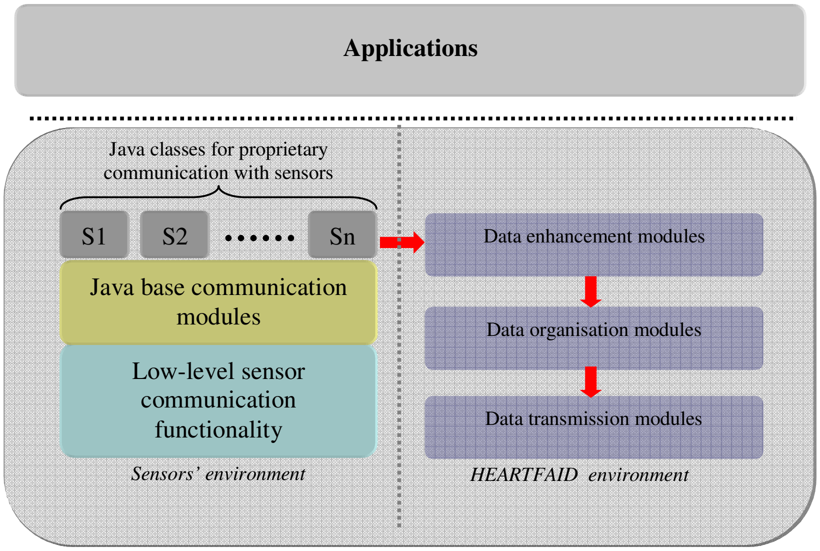
Applications
Java classes for proprietary communication with sensors
S1
S2
Sn
Java base communication modules
Low-level sensor communication functionality
Sensors' environment
Data enhancement modules
Data organisation modules
Data transmission modules
HEARTFAID environment

Figure 4.2 - Middleware in the home monitoring scenario

4.5 Needs and Requirements for the end-users' services interoperability

Users of HEARTFAID may be roughly classified into two "profiles": the medical personnel, and the patients, who have different, but equally important, aspirations for the platform.

In the following section we will give an overall view of the user requirements identified in relation to the project's interface. User needs are always difficult to capture and document, however we will try to provide an accurate analysis. Short descriptions of the two user profiles will be provided in the next section, followed by a list of user needs.

4.5.1 User Profiles

As stated above, following the initial requirements analysis of the HEARTFAID platform, two profiles have been identified (apart from administrative/technical staff) for the end-users: the medical personnel (administrators), and the patients (plain users).

a. Medical Personnel

The "Medical Personnel" profile is a wide-ranging category that includes all health-care professionals, involved with the research or the treatment of Heart Failure. The users associated with this category are the ones that will mostly interact with the system, and will make extensive use of the system resources.

b. Patients

The user-profile "Patients" describes all people suffering from Heart Failure, who are admitted into and monitored by HEARTFAID. Naturally, they are the ones, who will gain the most of the system, although they directly interact with it to a small extent.

4.5.2 User Needs

The requirements put forward from the prospective users, as regards the application interface, are many and quite demanding, as one may witness from the following list and description:

a. Completeness

The User Interface must provide tools to create, retrieve, update, and delete every piece of data in the platform, without the need for any external appendices, patches or complementary applications.

b. Accessibility

The full feature-set of the application should be reachable by all authorized users, through any contemporary PC or PDA – except, where specialized equipment is specifically requested.

c. Availability

The platform should be available to users at all hours, from any site, irrelevant of the remoteness of the location, the medium of communication, or any other external parameter.

d. Security

Authentication and Authorization are fundamental requirements, as are preserving the anonymity of all users, and the protection of their personal data.

e. Ease of Use

The UI should be designed according to the principals of Human-Computer Interaction, and following the paradigm of other popular computer applications, so as to ease its adaptation in the user environment.

f. Unambiguous Feedback

The reaction of the application to every user action should not bear any misinterpretation. As this is a medical application, this requirement is all the more essential and strict.

g. Consistency

The requirement for consistency also adheres to the HCI principals. In short, all graphical and textual elements of the UI should be similar across all components of the application.

h. Context Sensitive Help

The platform should provide an intelligible Help facility, to assist users with every feature of the interface.

i. Collaboration

Assist the interaction and co-operation of users.

j. Comprehensible Error Handling

In case an error in the application occurs, it should be handled gracefully, and provide meaningful feedback to the user.

The intricacy of the HEARTFAID platform and the diversity of user-application interaction that is required, make it especially challenging to satisfy all user needs. Nonetheless, all of the above requirements are considered equally important and great effort is invested on fulfilling them.

4.6 Clinical Standards

Interoperability is the ability of products, systems, or business processes to work together to accomplish a common task.

According to IEEE, interoperability is the ability of two or more systems or components to exchange information (functional interoperability) and to use the information (semantic interoperability) that has been exchanged.

With respect to ***software***, the term interoperability is used to describe the capability of different programs to exchange data via a common set of business procedures, and to read and write the same ***file formats*** and use the same ***protocols.*** Same meaning has to be applied with respect to ***medical devices.***

Thus, the issue of the identification of the most common standards for clinical data encoding and communication, is an ***interoperability*** issue.

In order to identify the most common interoperability standards for clinical data, it is necessary to analyse separately the medical devices/sensors and the health informatics products that have to co-exist in the HEARTFAID platform for the better management of the heart failure patients.

According to the European Directives currently in force:

- ❑ Active Implantable Medical Devices Directive 90/385/EEC
- ❑ Medical Devices Directive 93/42/EEC
- ❑ In-Vitro Medical Devices Directive 98/79/EC

A ***medical device*** is defined as any instrument, apparatus, appliance, material or other article, whether used alone or in combination, including software necessary for its proper application intended by the manufacturer to be used for human beings for the purpose of:

- ❑ diagnosis, prevention, monitoring, treatment or alleviation of disease,

- ❑ diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap,
- ❑ investigation, replacement or modification of the anatomy or of a physiological process,
- ❑ control of conception,

and which does not achieve its principal intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means.

A ***health informatics product*** is a software product for use in the health sector for health related purposes but excluding software which is

- ❑ Necessary for the proper application of a medical device or
- ❑ Is an accessory to a medical device or
- ❑ Is a medical device in its own right

Healthcare informatics products are increasingly being used within the healthcare sector, and, as a consequence, are becoming more intimately involved in patient care and healthcare professionals are becoming more dependent upon their use. For instance, these products can range from simple databases that are used to record and store medical data, to medical expert systems that are used to assist in the process of diagnosis of an illness.

4.6.1 Healthcare Informatics Products

In the world of information system the most diffuse standards in order to assure interoperability are CEN prEN13606/OpenEHR Archetypes (EHRcom), IHE RID (Retrieve Information for Display) and XDS (Cross-Enterprise Document Sharing), MML (Medical Markup Language), HL7 Messages (v2.x and/or v3) and HL7 CDA. An HL7 CDA document is a defined and complete information object that can exist outside of a messaging context and/or can be a MIME-encoded payload within an HL7 message; thus, the HL7 CDA complements HL7 messaging specifications. While HL7 CDA and MML only specify a content format but no communication protocol, RID and XDS only specify communication protocols and are “content agnostic”, that is, they do not define any content format. Only HL7 (with CDA) and EHRcom define both.

Considering their maturity and their adoption at worldwide level we consider that the most suitable interoperability standards are HL7 v2.x or v3 and CDA for the exchange of clinical documents. The name HL7 comes from “Health Level 7”, which refers to the top layer (level 7) of the Open Systems Interconnection (OSI) layer protocol for the health environment. HL7 is a standards-setting organization accredited by the American National Standards Institute (ANSI) and HL7 v2.x or v3 is a suitable standard for communication among the sub-systems of the platform or between the platform and an external third-party clinical system.

Hospitals and other health care centres or providers around the world require the ability to send and receive a vast amount of healthcare data, including patient information and various lab reports, on a daily basis. However, medical data can be extremely complicated due to the abundance of clinical terminology, as well as the structural complexity in the formation of the presented information. Thus, this information must be exchanged in a standardized format in order to ensure that the data is organized in a universally understood format. In order to achieve this, all healthcare information must be sent in a specialized healthcare language.

The language that has been developed to overcome these obstacles is HL7. The HL7 protocol consists of grammar and vocabulary that are standardized so that clinical data can be shared among healthcare systems and easily understood by all. By using the HL7 messaging protocol as a standard, all systems or sub-systems following the HL7 specifications are able to communicate easily with one another, without the need for information conversion.

The HL7 standard

HL7 has a message-oriented architecture. It means that the application in which an event occurs will send a message to other applications rather than serving a request.

HL7 messages are ASCII messages (unlike protocols such as DICOM), and the standard requires that they be "human readable". This is acceptable if you don't mind counting pipe characters. Messages are a defined sequence of segments and/or segment groups. Each message consists of the segments that are delimited by "carriage return" characters ("\r" or 0x0D). That's why you see each segment as a different line. Every line in a message is called a 'segment'. Each segment has its own semantic purpose. This means that it contains information of a specific type. Segments are the units that comprise a message. A segment is defined as a sequence of fields that also may or may not repeat. An HL7 message definition also states whether each segment is mandatory or not. Segments consist of fields that are composites. Composites are delimited by "|". Each field has its own unique purpose and is defined by the HL7 standard for each segment. Composites are the building blocks of segments. Composites may be either a primitive data type (string, number, etc.), or in turn be made up of other composites. Composites cannot have a recursive reference to themselves. The components of each composite are separated by ^ symbols, and the sub-components of these components themselves can be delimited using & characters.

In order to be as flexible as possible and achieve a consensus, the HL7 committees were forced to define a lot of fields as optional. The downside of this decision is that you cannot be certain that particular information will be present in a given message. This is one of the reasons why the same message may vary significantly from vendor to vendor. Thus, the use of optional fields should be agreed among the different sub-systems.

The messaging standard has been defined since version 2.x of the standard. Version 2.3 of the standard was approved as ANSI standard in 1997, version 2.4 in 2000 and version 2.5 in 2003 while version 3 is still work-in-progress.

Generically version 2.x has the following characteristics:

- ❑ Broad functional coverage
- ❑ Highly adaptable and flexible
- ❑ No reliable conformance tests of any vendor's implementation
- ❑ Vocabulary independent
- ❑ Weak technological base
    - ➢ not clear support for new technologies
        - ✓ Object Technologies
        - ✓ XML and Web Technologies
    - ➢ not clear support for security operations

Main features of version 3 are:

- ❑ Design based on *consensus* Reference Information Model (Model Driven Architecture)
- ❑ Entire health and clinical management domain
- ❑ Vocabulary-level interoperability
- ❑ Explicit conformance model
- ❑ Adaptable to current and future technology bases
- ❑ Built on strongly accepted industry base technologies

A major decision that will need to be taken is the choice of the HL7 version that will be implemented in all the various sub-systems of the HEARTFAID platform.

The HL7 Clinical Document Architecture (CDA)

The HL7 Clinical Document Architecture (CDA) is an XML-based document mark-up standard that specifies the structure and semantics of clinical documents for the purpose of exchange. CDA provides an exchange model for clinical documents, such as discharge summaries and progress notes, and brings the healthcare industry closer to the realization of an electronic medical record. By leveraging the use of XML, the HL7 Reference Information Model (RIM) and coded vocabularies, the CDA makes documents both machine-readable (so they can be easily parsed and processed electronically) and human-readable (so they can be easily retrieved and used by the people who need them).

A clinical document contains observations and services and has the following characteristics:

- ❑ Persistence – A clinical document continues to exist in an unaltered state, for a time period defined by local and regulatory requirements.

- ❑ Stewardship – A clinical document is maintained by a person or organization entrusted with its care.
- ❑ Potential for authentication - A clinical document is an assemblage of information that is intended to be legally authenticated.
- ❑ Wholeness - Authentication of a clinical document applies to the whole and does not apply to portions of the document without the full context of the document.
- ❑ Human readability – A clinical document is human readable.

A CDA document is a defined and complete information object that can include text, images, sounds, and other multimedia content. It is based on RIM and on HL7 v3 data types and it has been created with the aim of standardizing the organization and structure of medical documents and easing the exchange of documents.

A CDA document is comprised of a header, referred to as the "CDA Header" and a body referred to as the "CDA Level One Body".

A CDA document is wrapped by the <ClinicalDocument> element, and contains a header and a body. The header (constant across all CDA documents) lies between the <ClinicalDocument> and the <StructuredBody> elements, and identifies and classifies the document and provides information on authentication, the encounter, the patient, and the involved providers.

In their header, CDA documents can reflect unauthenticated, authenticated, or legally authenticated state (indicated by <authenticator.type_cd> and <legal_authenticator.type_cd>). No provision for electronic signature, but acquisition of signature is documented by <signature_cd>.

The CDA level one body consists of either nested containers (sections, paragraphs, lists, tables) or a non-XML element. The <non-XML> element:

- ❑ Represents a document body that is in some format other than XML
- ❑ Uses Encoded Data (ED) data type
- ❑ Used only to reference data stored externally to the CDA Level One document

The body contains the clinical report, and can be either an unstructured blob, or can be comprised of structured markup. A CDA document section is wrapped by the <Section> element. Each section can contain a single narrative block and any number of CDA entries and external references. The CDA narrative block is wrapped by the <text> element within the <Section> element, and provides a slot for the human readable content needing to be rendered. Within a document section, the narrative block represents content to be rendered, whereas CDA entries represent structured content provided for a computer.

Example sections include: History of Present Illness, Past Medical History, Medications, Allergies and Adverse Reactions, Family History, Social History, Physical Exam [Vital Signs, Skin, Lungs, Cardiac], Labs, In-office Procedure, Assessment, Plan, etc.

The CDA specification defines a multi level architecture where each level is derived from a more basic level. Levels refer to varying degrees of required markup granularity and RIM-derived markup, but clinical content remains constant, regardless of the extent of added markup.

Three different levels are available and levels establish baselines for conformance claims.

- ❑ Level One -- RIM-derived document header. Body is largely structural, although codes can be inserted.
- ❑ Level Two -- HL7 Templates can constrain the general Level One DTD, resulting in Level Two DTDs.
- ❑ Level Three -- Clinical content can be marked up to the extent that it is modeled in the RIM.

A CDA document is a defined and complete information object that can exist outside of a messaging context, complementing HL7 messaging specs. CDA documents can be encapsulated as MIME packages within HL7 messages.

4.6.2 Medical Devices

The need is for intercommunication among medical devices and clinical information systems. Infusion pumps and ventilators commonly have RS-232 ports, and these devices can communicate with many physiological monitoring instruments. Products to link medical equipment and personal communication devices exist as well. However, virtually all of these are specialized applications—custom interfaces unique to the two devices being linked. The fact that an infusion pump from Company A can communicate with a patient monitor from Company B does not guarantee that Company A's pump can communicate with the same type of monitor from Company C. Interfacing two devices with "standard" RS-232 ports does not ensure communication, because there are many different ways to send data over that serial interface. Matching the connectors and pins can be problematic, as is establishing a handshake. Moreover, medical device design is not perfected simply because data can be sent from one device to another; the devices must be able to understand the format and content of the messages they communicate to each other. They must speak the same language, both grammatically and semantically.

The use of standard can facilitate the attainment of interoperability, thus while interoperability is the goal, standards are the ways.

Several standards are available in order to assure interoperability in medical device communication and data format. Medical devices can be divided into two categories:

- ❑ Imaging Medical Devices
- ❑ Non-Imaging Medical Devices

Imaging Medical Devices

Digital **I**maging and **Co**mmunications in **M**edicine (DICOM) is a comprehensive set of standards for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol.

The communication protocol is an application protocol that uses TCP/IP to communicate between systems, known as Application Entities. DICOM files can be exchanged between two entities that are capable of communicating using DICOM transfer protocol. Each DICOM file may include image (i.e., Computed Tomography, Magnetic Resonance Imaging), video (i.e., Ultrasound) or waveform (i.e., Echocardiography) and patient related data.

DICOM enables the integration of scanners, servers, workstations, printers, and network hardware from multiple compliant vendors. In order to ensure compatibility among different vendors and machines a well defined document that includes DICOM conformance statements is provided. The latter clearly state the DICOM functions supported from each Application Entity.

DICOM is also particularly important in the context of the Integrating the Healthcare Enterprise initiative (IHE), which employs existing standards, such as DICOM and HL7 (as mentioned earlier), for developing efficient and seamless workflows (i.e. integration profiles) using medical equipment and information systems from multiple vendors. The only drawback of DICOM is the complexity of the standard that requires a developer to have a prior knowledge of DICOM philosophy. However, DICOM is currently the most widely adopted protocol by hospitals and smaller but distributed specialized centers related to the health sector.

Non-Imaging Medical Devices

In the field of non-imaging medical devices, standards are not largely adopted by manufacturers. On the other side, the integration of medical devices with proprietary protocol/data format is time-consuming and not always feasible (several times manufacturers do not disclose the necessary technical information). In order to perform the integration of a medical device with proprietary format, there are the following possibilities:

- ❑ The release of technical information by the manufacturer (usually under signing a NDA).
- ❑ The purchase of some additional modules from the manufacturer that allow the communication with the medical device and the conversion or export of the examination file in a standard or "disclosed" format.

And this work has to be done differently for each different medical device with proprietary protocol/data format.

Considering these problems, IEEE decided to approach this issue systematically. To address the medical device plug-and-play interoperability problem, a single communications standard is needed. Software engineers designing medical equipment could use such a standard to implement external interfaces once for all models.

In the IEEE 1073 / ISO 11073 Series of Standards the concept of Point-of-Care medical device communication is described:

- ❑ Provide real-time plug-and-play interoperability for patient-connected medical devices
- ❑ Facilitate the efficient exchange of vital signs and medical device data, acquired at the point-of-care, in all health care environments

Leveraging off-the-shelf technologies, scaling across a wide range of system complexities, and supporting commercially viable implementations

These standards, intended to bring a wide range of medical devices under their purview, aim to encompass transparent plug-and-play interoperability, ease of reconfiguration, and ease of use.

The medical device communications problem has three principal parts:

- ❑ lower-layer, or transport, services
- ❑ upper-layer application profiles
- ❑ upper-layer semantic, or device-specific, object data models

For each part, the technology that best fills the needs of a given medical device and host system can be selected without having a major effect on the other two parts. The IEEE 1073 standard set generally reflects the tripartite structure of the medical device intercommunication challenge. It consists of a base standard, which provides an overview and framework for the family, and the following focused standards.

Unfortunately this well-structured family of standards is not largely adopted by manufacturers yet.

There are other standards (de jure or de facto) for data communication and data format related to specific medical devices. From an analysis of deliverable D5 it seems that in HEARTFAID the most used (in routine workflows (workflows 1 and 2 of D5)) non-imaging medical devices will be "Resting ECG" and "Holter ECG".

Resting ECG Medical Devices

The most common examination is the Resting ECG where the ECG in supine position is acquired with 12 lead (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6) for a short time (usually 10 sec).

In the domain of electrocardiology, the OpenECG project has made a huge effort in the attempt of promoting interoperability. Several tools and documentation are available in the OpenECG portal related to interoperability standards in the ECG domain.

The main standards for resting ECG interoperability are:

❑ SCP-ECG (EN 1064:2005)
❑ HL7 Annotated ECG

Since its first publication in 1993, the SCP-ECG standard was further reviewed and published (in its last version 2.1) as an official CEN standard in March 2005 (EN 1064:2005).

The ***SCP-ECG standard*** relates to the conventional recording of the electrocardiogram (i.e. the so-called standard 12-lead ECG and the vectorcardiogram (VCG)). It is specific for: "Short-term conventional ECG (Resting ECG)".

The SCP-ECG standards relates to:

❑ Data encoding, transmission and storage of short-term resting ECG.
❑ Two-way digital transmission of remote requests and results between digital electrocardiographs (ECG carts) and heterogeneous computer systems (hosts) or computer ECG management systems.

In SCP-ECG the record is the entire data file (binary) which has to be transmitted, including the ECG data and associated information, such as patient demographics and other clinical data.

The record is composed by checksum (CRC), length of the record and data record that is divided into different sections. Some sections are mandatory, others are optional (depending on the selected compliance category), others are manufacturer-defined.

The compliance categories provide users and manufacturers of ECG devices and/or systems with a relatively simple codification of SCP-ECG related features and information content that may be provided by a specific device.

The SCP-ECG record is a binary file but it has the advantage that ECG viewers are freely available in the OpenECG site and that the high-compression mechanism implemented in the standard allows compression factors of almost 20.

The ***HL7 Annotated ECG*** (HL7 aECG) standard is more recent and it has been developed to answer the following issue:

- ❑ New Drug Application sponsors collect biological data, often as waveforms from subjects dosed with the candidate drug.
- ❑ A number of measurements are made from the data, or from close derivations of that data. Those measurements are compiled into datasets and statistically analyzed.
- ❑ The datasets are submitted to FDA by the New Drug Application sponsors to support the findings.
- ❑ The FDA would like to get a sense of the accuracy and consistency of the measurements made from the collected biological data.

Thus, the goals of such a standard were:

- ❑ To facilitate the submission of the biological data or close derivations of it used to make the measurements.
- ❑ To allow the annotation of the biological data with points and intervals to show the reviewer relevant landmarks used for making the measurements.

Main features of this standard are:

- ❑ Based on XML
- ❑ Sponsored by the main manufacturers (GE, Philips, Mortara, etc.)
- ❑ ECG compression is not supported by the standard
- ❑ File size is significantly higher than SCP-ECG
- ❑ HL7 aECG is less “mature” than SCP-ECG
- ❑ Manufacturers implementation of HL7 aECG is often “not-interoperable”

Holter ECG Medical Devices

The most common examination is the 24-hour Holter ECG where the ECG during the patient daily life is acquired usually with 3 leads and with a sampling rate from 125 to 250 Hz. It is also called “electrocardiographic ambulatory Holter monitoring” (AECG).

Usually the reading station (a PC) reads the data acquired with the data logger device and, after interaction with the reviewer cardiologist, produces a report. The reading station is usually capable to export the examination and the most common disclosed formats are:

- ❑ MIT/BIH
- ❑ ISHNE (uncompressed)
- ❑ EDF or EDF+

Where the raw data is entirely available with some additional information.

The ***MIT-BIH format*** is promoted by MIT and PhysioNet.

In this format, a record consists of at least three files, which are named using the record name followed by distinct suffixes (extensions) that indicate their contents (.atr, .dat, .hea). The .dat (signal) file contains digitized samples of one or more signals. It can be very large. The .hea (header) file is a short text file that describes the signals (including the name or URL of the signal file, storage format, number and type of signals, sampling frequency, calibration data, digitizer characteristics, record duration and starting time). Records can include a binary *annotation* files (.atr). Annotation files contain sets of labels (annotations), each of which describes a feature of one or more signals at a specified time in the record (i.e. an annotation for each QRS complex (heart beat) in the recording, indicating its location (time of occurrence) and type (normal, ventricular ectopic, etc.), as well as other annotations that indicate changes in the predominant cardiac rhythm and in the signal quality).

The ***ISHNE format*** is promoted by ISHNE (International Society for Holter and Non-invasive Electrocardiology).

In this format, a record consists of a file that contains magic number, checksum, a header (lead specification, lead quality, amplitude resolution, etc.) and ECG data (information about the storage of the ECG, information about the patient, lead samples, etc.).

The ***EDF or EDF+ format*** was defined by a European group of researchers originally for sleep analysis, but it is usable for any series of multiparametric data.

One data file contains one uninterrupted digitized polygraphic recording. A data file consists of a header record followed by data records. The variable-length header record identifies the patient and specifies the technical characteristics of the recorded signals. The data records contain consecutive fixed-duration epochs of the polygraphic recording.

4.7 Bibliography and References

[1] http://www.openclinical.org

[2] http://www.e-europestandards.org/activities_health_online.htm

[3] Eichelberg M, Aden T, Riesmeier J, Dogac A, and Laleci GB. "ELECTRONIC HEALTH RECORD STANDARDS – A BRIEF OVERVIEW". ITI 4th International Conference on Information & Communications Technology 2006 (ICICT 2006).

[4] Chronaki C, Chiarugi F, and Reynolds M. "Grid-enabled medical devices, innovation in eHealth, and the OpenECG network". Proceedings of ITAB 2006.

[5] http://www.interfaceware.com/manual/ch-2.html

[6] http://www.hl7.org/

[7] http://www.kestral.com.au/hl7v3/resources

[8] https://www.hl7.org/library/data-model/RIM/C30202/rim.htm

[9] http://xml.coverpages.org/ni2004-08-20-a.html

[10] http://www.cs.unicam.it/ontogov06/public/2006/ontogov06_Marcheschi.pdf

[11] http://www.e-osiris.it/documenti/s_HL7/HL7_CDA_ProgramOverview.htm

[12] http://www.srdc.metu.edu.tr/webpage/seminars/Healthcare/Clinical%20Document%20Architecture.ppt

[13] http://medical.nema.org/

[14] http://www.ewh.ieee.org/r8/ukri/embs/Downloads/IEEE1073_Standards_Abstract.pdf

[15] http://www.ieee1073.org/standards/1073standards.html

[16] http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37890

[17] Chiarugi F, Lees PJ, Chronaki CE, Tsiknakis M, and Orphanoudakis SC. "Developing Manufacturer-Independent Components For ECG Viewing And For Data Exchange With ECG Devices: Can The SCP-ECG Standard Help?", Proceedings of Computers in cardiology 2001;28:189-192.

[18] Zywietz C, Fischer R, and Joseph C. "Communication and Storage of Compressed Resting and Exercise ECGs Using the Revised SCP-ECG Standard", Proceedings of Computers in cardiology 2001;28:189-192.

[19] Sakkalis V, Chiarugi F, Kostomanolakis S, Chronaki CE, Tsiknakis M, and Orphanoudakis SC. "A Gateway between the SCP-ECG and the DICOM 3.0 Supplement 30 Waveform Standards", in Proceedings of Computers in Cardiology 2003;30:25-8.

[20] Fischer R, et al. "Communication and Retrieval of ECG Data: How Many Standards Do We Need?", Computers in Cardiology, 2003;30:21- 24.

[21] Willems JL, et al. "Common standards for quantitative electrocardiography: goals and main results. CSE Working Party", Methods Inf Med. 1990 Sep;29(4):263-71.

[22] Health informatics - Standard communication protocol - Computer-assisted electrocardiography. ICS: 35.240.80. IT applications in health care technology, Reference number EN 1064:2005. http://www.cenorm.org.

[23] Reynolds M, Norgall T, and Cooper T. "State of the art technology – Protocols, Networks and Standards for point-of-care device communication, CEN/ISO/IEEE 11073 Standards", WSC Workshop, Geneva, 26-27 February, 2004.

[24] The OpenECG portal. http://www.openecg.net.

[25] How to Implement SCP – Part I. http://www.openecg.net/tutorial1/index.html.

[26] How to Implement SCP – Part II. http://www.openecg.net/tutorial2/index.html.

[27] Brown B, Kohls M, and Stockbridge N. "FDA XML Data Format Design Specification". http://www.openecg.net/member/FDA_HL7/FDA_20_XML_Data_Format_Design_Specification_DRAFT_C.pdf (access reserved to OpenECG members).

[28] Morganroth J. "Ambulatory Holter electrocardiography: choice of technologies and clinical uses", Annals of internal medicine (Ann. intern. med.) 1985, vol. 102, no1, pp. 73-81.

[29] Scientific data formats. http://www.dpmi.tugraz.at/schloegl/matlab/eeg/.

[30] An Introduction to the PhysioBank Archives. http://www.physionet.org/physiobank/physiobank-intro.shtml#records".

[31] Zareba W, Locati EH, and Maison Blanche P. "The ISHNE holter standard output file format: A step toward compatibility of holter systems", Annals of noninvasive electrocardiology (Ann. noninvasive electrocardiol.) 1998, vol. 3 (1), no3, pp. 261-262.

[32] Kemp B, Värri A, Rosa AC, Nielsen KD, and Gade J. "A simple format for exchange of digitized polygraphic recordings", Electroencephalography and Clinical Neurophysiology, 82 (1992) 391-393.

5. Functional Specifications of the Middleware

5.1 Functional Specifications of the integration/interoperability middleware

The Middleware will be composed of the following main components:

- Resource Management
- Component Interfaces
- Architectures and Data Models

Resource Management Systems (RMS) have a main role in handling a dynamically evolving set of distributed, interconnected and heterogeneous hardware devices. In the next section we will show some common implementation patterns and some example of RMS taken mainly from the GRID computing context where aspects such as security, fault tolerance, autonomy and heterogeneity have been faced intensively.

Component interfaces are not trivial aspects when designing a complex system such as the HFP. In particular, when the systems are logically and physically distributed across different computing resources, a set of common features to be faced are communication, protocols, automation/interaction techniques, dynamic service discovery.

Architectures and Data Models refer to the Remote Data Acquisition (RDA) level of the HFP, which is with the low level Integration Middleware and interface data acquisition networks mainly composed of remote sensors and measurements devices.

5.1.1 Resource Management

Resource Management is a complex task involving security, fault tolerance, scheduling, flexibility, extensibility and naming of resources.

In the viewpoint of the HF platform, the most important features of a possible implementation of a RMS are fault tolerance, flexibility, extensibility and naming of the resources.

Flexibility and extensibility are achieved by applying appropriate Resource Dissemination and Resource Discovery mechanisms.

An ad hoc Resource Dissemination Protocol can be used to *advertise* a resource in the framework and to report the status of a resource to a RMS. Protocols for Resource Dissemination can be:

- **periodical**: in periodical dissemination protocols status information of a resource is batched and reported periodically to the RMS. This information can be pulled (it is the RMS that collects the information from the resources) or pushed (the resources send status information to the RMS).
- **on demand**: resource information is updated when a specific event or status change triggers the update.

Resource Discovery is tightly bound to Resource Dissemination and can be implemented through:

- **Query based** approaches where the system is queried by the RMS to find out availability of resources.
- **Agent based** approaches where discovery agents traverse the system to gather information about resource availability.

A RMS should be able to identify every resource used in the platform. To do this a common and *global name spacing* mechanism will be required. Global name spaces can be organised in three different policies:

- **flat**: in this scheme every resource gets a name at creation time and this name carries no information on the structuring imposed by the system architecture. Flat name spaces are not very scalable and thus they are not quite suitable for the framework.
- **hierarchical**: with this policy the layer structure of a system architecture is transported to the naming convention used for the resources. Resource names are constructed by traversing the hierarchical structure in a top-down visit.
- **graph based**: with this policy no hierarchical ordering is established thus resources are named by following the links from one object to the other in the graph structure.

In the HF context, the RMS will have a central role in the correct functioning of the platform of services.

Fundamental requirements of the RMS will be flexibility and adaptability. In fact, the platform must be able from one side to interact with existing devices, EPR, HIS and so on, outside the HF world, and from the other side it should guarantee a high level of scalability as well as the possibility to be easily extended with new modules and functionalities without having to re-engineer the entire architecture.

These goals will be achieved through the implementation of widespread standards at different levels of the platform architecture: for data encoding, for information exchanging, for documents/reports production, for data recording and warehousing, and of course for resource encoding and management.

It will be challenging to enable the extension of the platform with new software modules in a plug-n-play fashion, thus achieving a very high level of flexibility and extensibility.

In addition to these aspects, the RMS should guarantee security and confidentiality of data, that is to ensure that only authorised users are able to access data (entirely or part of it), accessed data are not altered somehow (integrity), the identity of patients is safeguarded, due authorisations have been requested and obtained when necessary, and source measurements are recorded without unwanted interferences or alterations.

5.1.2 Component Interfaces

Implementing clean interfaces to connect software components is a quite important task in a project that presents modularity, scalability, extensibility and generality as its main features. Moreover, in many cases the project encompasses distributed computing resources, be it software like processes and data or hardware resources like embedded micro or nanodevices, gateways, servers or storage devices. In these scenarios the design complexity for component interfaces grows rapidly involving mechanism for interprocess communication, design of information exchange protocols, remote service requests and remote service discovery, synchronization, handling of heterogeneity.

There is a great number of possible comparison arguments for studying component interfacing software. Possible choices are:

- **Data Representation**: What kind of data is moved and how it is wrapped to preserve typing information among possibly heterogeneous components.
- **Transport Protocol**: How data is moved across the connection that binds the (possibly distributed) components.
- **Service Description** and **Service Location**: How services available on servers are described and if such a description exists for a specific implementation. What kind of formalism can be used for discovery of service availability?
- **Language Binding** and **Language Paradigm**: In which programming language the formalism can be embedded and what kind of paradigm does it conform to (object oriented, functional, streaming)?
- **Remote Reference**: What is the representation of a reference to a remote service supplier component?
- **Synchronisation**: Whether the communication underlying the component interaction is implemented in a synchronous or asynchronous way.

A very important note is that many bridge middleware tools exist that enable components using different architectures for their interfaces to transparently communicate each with the other.

As a final remark, it should be noted that often the only way to interconnect components that are distributed on an open network such as the Internet, is to encapsulate all information into HTTP packets. This technique is called tunnelling and it is useful to bypass security firewalls that often enable only HTTP communications to access the target server machine.

In the HF context the aspects of **Service Description** and **Service Location** are particularly important due to the flexibility and extensibility aspects which we would like the platform to have. In particular, the middleware shall implement adequate mechanisms to allow both existing and new components to expose specific services that other components or users are able to locate and exploit. This implies that within the middleware a specific language should be adopted that all the components shall necessarily speak. Of course, standard approaches will be investigated for two important reasons: 1) to get sufficient support from the scientific community that is experimenting that standard; 2) to increment the possibility to integrate components that have already been developed and tested by third parties.

It is worth making a short comment to the **Synchronisation** aspect. Due to the large variety of services offered to the end-users, as well as of the software components that will be integrated into platform, it will be necessary to implement different kind of synchronisation approaches according to the urgency/severity of the operation to be performed and, if necessary, also on a priority basis. In this last case, adequate mechanisms for priority management shall be investigated and implemented; standard solutions will be adopted since it is not interest of this study to experiment new approaches.

5.1.3 Architectures and Data Models

This section deals in particular with the RDA level of the HFP, which is with the low level Integration Middleware.

A great attention, in commercial and academic environments, has been directed toward building data acquisition networks mainly composed of remote sensors and measurements devices. The main differences between sensor-based data sources and other (often well-engineered) business-like data sources introduce some problems when trying to adapt traditional database approaches. On the server side it is necessary to design architectures that can access and efficiently query sensor data stream without wasting the limited energy resources on the sensors themselves. On the client-side, the software installed on the acquisition devices can use the available computing power to partially execute database operations like joins or aggregations within the sensor network reducing the very time and energy consuming communication operations.

The following aspects should be taken into account when acquiring data from a RDA network, according to the specific activity performed by the platform:

- **Near real-time processing** because it might be too expensive to store whole data streams to disc space and because sensor streams represent real world events (accidents, failures) which have to be faced in real-time.
- Sensors are typically **not reliable**.
- Sensors have own **computing power** which can be used to partially compute queries as data flows through them.

Server-Side Issues

A first issue related to traditional database behaviour that has to be enhanced to face RDA networks requirements is the assumption of static data sources. This assumption does not apply to acquisition devices that provide, instead, real-time (or delayed) data. In particular, the most complex and challenging scenario is the acquisition of a streaming source of data. The case of store&forward devices (or intermittent acquisition devices) can be considered as a particular case of the continuous streaming.

The major problem is that a query processor cannot wait indefinitely for data requested to devices that can be temporarily down because of low power, or disconnected due to particular environmental conditions or maintenance operations.

Possible solutions are shown in Volcano and Fjords (see references) and focus on the introduction of *queues* which present different kind of asynchrony mechanism and pull/push strategies.
Another important issue is related to ensure maximum resource exploitation and optimization.
Reusing data produced and sent by acquisition devices on the central query processor to answer sets of related (overlapping) queries has both the advantages of limiting storage space required in the database and reducing the number of times a sensor has to deliver the same data for different queries.
Adjusting the sample rate can be very useful for reducing the overhead introduced by acquisition devices that go on sampling at high frequencies when it is not required.
Directing the acquisition devices to aggregate samples in predefined ways or to drop sensed values that exceed some threshold values can reduce the quantity of information that travels inside and outside the HFP.

Other issues have to be faced when the discussion moves from a data acquisition viewpoint to a viewpoint more related to querying the data once it has been received.
Continuous Queries (CQ) is a mechanism that allows a user or software module to query data that is continuously updated over time. A common technique to approach CQ is to store, on the central query processor, the posted queries. When a tuple arrives from the sensors, the related queries are applied to it and the results are forwarded to the poster of the query.
Some experiments have been made to extend the CQ concept to be adaptive and to apply also to streamed data. Adaptivity refers to changes in the query set, query status, sensors, sensor sample rate or operator order. Applicability to streams requires the query operators to be non-blocking and the introduction of new operators for some common operation that do not match easily the stream concept (for example the sort operation).

Example Data Model and Architectures

The Data Model at the application level used for mapping RDA networks is a very important issue when designing the Middleware of the HFP. It has to be general enough to encompass sensible modifications to the RDA network (topology, number of devices, type of sensors), to a single acquisition device (geometry, substitution, removal or upgrade) or to its behaviour (scaling of measures, sample rate, aggregation of operations). At the same time it should enable users to navigate the RDA network by using queries in an intuitive and efficient way.
A quite interesting example of how data can be managed in an object oriented way is shown in Figure 5.1 (see references). The class diagram shows the component classes that are used to model a general system of observations and measures.

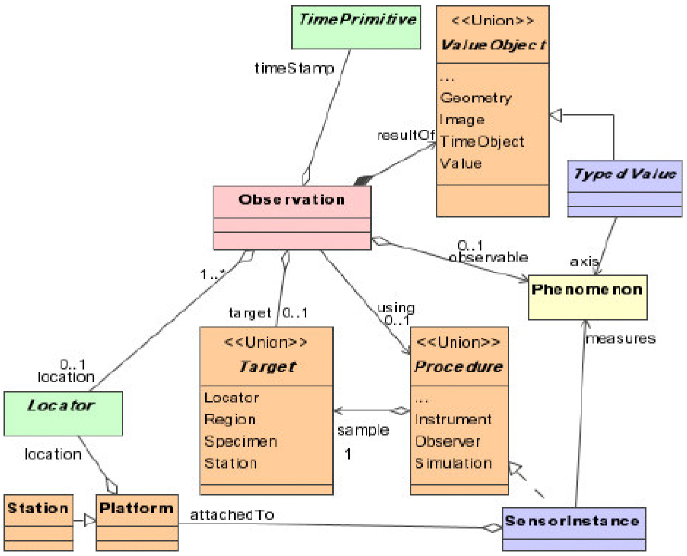

Figure 5.1 - Class diagram of the object model proposed by the Open GIS Consortium

Another quite interesting project related to Sensor network is Eyes (see references). In Eyes the system view is focused on a layered architecture, based on well defined interfaces, that matches those of operating systems and network stacks. Whilst in the Open GIS Consortium proposal the key point is representation of the data at a higher level, in Eyes the focus is on how data is gathered in the RDA network and how it is forwarded to the higher application layers emerging along the system architecture layers. An example illustrating the reference architecture is shown in Figure 5.2. Applications can interface a distributed system layer that shows the underlying network as a set of distributed systems who offer some services. The *Lookup Component* is used for getting information on the network and to query capabilities related to the services. The *Information Component* is accessed for gathering the information produced by the sensor network.

The lower layer is called Operating System (OS) layer and offers the implementation of the services exported by the Distributed Services layer.

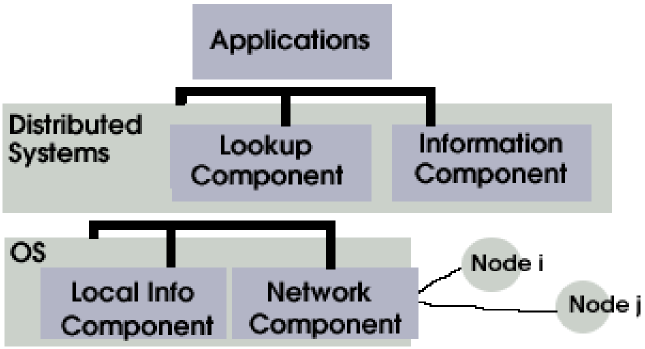

Figure 5.2 - Eyes layered architecture

These examples can be used as reference solutions for the implementation of the HF middleware, especially the Integration Middleware that will be in charge of managing both the RDA network and the other modules of the platform that need to access data.

5.2 Functional specification of HF end-user applications

Under the broad term "End-user Services" are implied a number of different components, which capacitate access to the application's utilities. As stated above, HEARTFAID encompasses many different processing modules, which all require means of effective, bi-directional communication with the users. The challenge, of providing this communication successfully, lies in identifying user needs, determining functional specifications, and finally designing and implementing a comprehensive and convenient User Interface to address them. This section attempts to provide an accurate description of the End-User service components and to outline the functional specifications for each one. It is divided into three main sections, which correspond to the three main subsets of End-User services: patient telemonitoring, web-based portal, alert and alarm service.

5.2.1 Patient Telemonitoring

Heart Failure patients need to be monitored by physicians in a very frequent basis. Remote monitoring can help health providers to develop a home monitoring program for such patients, with a great saving of resource and a better service for the patients. The interface to this program is the "Telemonitoring module", which is divided into client- and server-side components. The module provides significant benefits in the patient's quality of life and quality of treatment, and is presented below, in detail.

a. Client-side Component

Each patient that participates in HEARTFAID should be able to privately record and send to the system accurate readings of vital signs, biometrics, and any other datum deemed important. To achieve this task, a user needs:

i. Sensor device(s), to make the readings,
ii. A PC or PDA, that will retrieve the data from the sensor, and
iii. Access to the Internet, to send the collected data to the HEARTFAID system.
Whether the patient uses a PDA or PC, she will follow the same simple steps to complete the process. The User Interface for this module will be designed as straightforward as possible, and will guide the user through the process with short, easy instructions. The control flow, for the client-side component will be:

- The software initiates – either on PDA or PC – and detects the sensor device(s), and the available Internet connection. If the detection is successful, a message appears to inform the user, and ask her to proceed with the readings.
- Once the readings are complete, the module bundles them together, and uses the Internet connection to access the Telemonitoring Server, and send the data bundle. Alternatively, if a connection is not available, the module could save the data, and send them at a later time, when a connection becomes available. This case is considered especially useful for mobile users, for whom the availability of an Internet connection is less probable.
- The module uses a synchronous communication protocol when accessing the Telemonitoring server, and awaits an acknowledgment of successful delivery, after sending the data bundle. If such an acknowledgment is received, the module presents a confirmation message to the user. If a negative acknowledgment is received, or a time-out occurs, the module re-starts the process.

b. Server-side Component

The other integral component of this module is the Telemonitoring Server, which listens for requests from the remote clients. The server implementation needs no specialized sensors or other equipment; server hardware, a broadband connection, and a port open to the Internet suffice. In the server-side, the flow of control follows these steps:

- The component should be available “24*7”, to listen for requests from the client PDAs and PCs. When a request is received, some limited processing takes place, to ensure the data bundle is complete and not corrupt. If these simple requirements are fulfilled, the server sends an acknowledgment of successful delivery to the client. If not, the server sends a negative acknowledgment, to indicate the need to retry.
- Once the data is correctly received, the server saves a local copy, for backup and logging purposes. The data bundle is then transformed to an XML document, consistent with the Clinical Document Architecture (CDA) standards, and sent to the central database of the middleware platform.

5.2.2 Web-based Portal

The main volume of work, of the End-User services, will be implemented for the Web-based portal. Users of this service are the medical personnel, i.e. doctors, nurses, the hospital's administrative personnel, and research scientists. The portal is in effect the doorway to a multitude of tasks and utilities offered by the middleware platform. The principal objectives of this module are to:

i. satisfy all user requirements,
ii. limit as much as possible the complexity of the interface,
iii. provide a single access point for communicating with the application
iv. ensure uniformity of all portal features, such as information display, error-handling, feedback, etc.

The core components of the portal are the Electronic Health Record (EHR), the Personnel Management, and the Decision Support System (DSS) Administrator. Extensive portrayals of each portal component are presented below.

a. Electronic Patient Record (EPR)

The EPR provides the interface for the Patient-Data Management module. The Use Cases that should be covered within this front-end module are:

- Create a new EPR for a patient. The interface should enable the user to enter all relevant information for a new patient, including demographic data, medical history, family history, and a record of medical encounters.
- Retrieve and View patients' EPRs. The interface should support Search functionality for EPR repository, both for single- and multi-criteria queries. The user should also be able to review the patients' records, either in a list or individually.
- Update patients' EPRs. The user should be able to make corrections and additions to the patients' electronic records, through this web-based interface.
- Delete patients' EPRs. Another important attribute of this front-end module is the ability to delete any of the stored electronic records, either in a batch – e.g. multiple selection from a list – or individually.

b. Personnel Management

Fundamentally, this module encapsulates the interface to the User Management functionality. The applicable Use Cases for this module are:

- Create a new personnel record. The interface should be designed to ease the entry of a new staff-member in the repository, by presenting clear and straightforward access to this function.
- Retrieve and View personnel records. The user should be able to query the staff repository, with any number of applicable criteria, and view a comprehensible list of results. Furthermore, the user should be able to select and view individual items from the list.
- Update personnel records. This is also an integral feature of the front-end. The interface should entail an easy way of updating or appending data in a personnel record.
- Delete personnel records. Finally, besides creating and updating, the user should have in her disposal a way of deleting records from the staff repository, both individually and in a group.

c. Decision Support System (DSS) Administrator

The Decision Support System is the core of this project, and provides indispensable services to the care takers. The interface to this module is the most demanding aspect of the End-User services, as it may even affect the frequency of use, and the effectiveness of the module. There are quite a few Use Cases that must be supported, namely:

- Retrieve and View Processing algorithms. The DSS entails various algorithms for data manipulation. The interface should offer to the user the ability to view a listing, as well as the detailed settings of each algorithm.
- Update Settings of Processing algorithms. Besides simply viewing the supported algorithms, users should be capable of choosing and modifying their settings, in order to fine-tune the processing.
- Create new Rule. The user may also influence the results of the DSS processing, by applying new rules and rule sets. The interface should offer a simple and manageable way to carry out this function.
- Retrieve and View Rules. Any rule that is entered into the persistent repository should be accessible via an unambiguous interface, where the user may perform a simple or complex search, and view all matching results.
- Update Rules. Just like with any other information that is submitted to the application, rules should be easily modifiable.
- Delete Rules. The portal should also provide means for the removal of unwanted or obsolete items from the rules repository.
- Run Processing algorithm on test set. Executing a processing sequence is the most important part of the DSS. The portal should facilitate it, with a user-friendly interface for selecting a test set, selecting one or more algorithms and rules, executing the processing, and presenting the results in human readable format. It may be necessary to present results in a multitude of formats, for viewing, storing, sending, further processing, etc.
- Request Inference. The inference request for a particular patient is also an integral part of HEARTFAID. In this case, the interface should allow the selection of a particular patient – includes a previously specified Use Case, "Retrieve and View patients' EHRs" – selection of one or more algorithms and rules, and commencement of execution. The results should be displayed in a coherent and well-documented fashion, to present the reasoning as well as the outcome of the inference.

5.2.3 Alert and Alarm service

The main objective of this service is to provide advanced alert and notification communication services to HEARTFAID users.

Within this subproject, a number of different information delivery methods will be examined in order to study and develop a HEARTFAID platform interface dedicated to mobile devices (mainly mobile phones and PDAs). Using this interface, a number of HEARTFAID medical services will be available over GSM network to mobile users.

Emphasis will be given to alert and notification services for both medical staff and patients. In the framework of this task, the instant communication method of Short Messaging System (SMS) will be used in order to provide HEARTFAID platform with enhanced one and two-way communication services available for mobile users.

The key issues for this service will be the advanced user profiling and the cognitive techniques which should be used in order to dynamically compose and send alert and notification messages to the users, depending on their particular personal profile, according to user's attributes (doctors or patients).

Needs analysis about registration and profiling services will be based on registration and authentication services, according to each user's profile, that will be used to implement a personalized messaging platform. Personalised access technologies to the HEARTFAID platform will also be brought in focus to ensure that medical professionals and patients will have easy and secure access to all the necessary information.

As stated above, the main communication medium will be text mobile messaging service delivered to personal mobile phones in order also to address the patient confidentiality matters.

In general, SMSs are sent as:

- Reminders: This function provides the transmission of reminder SMSs from the HFP to the mobile users. These text messages are focused on reminding the patients about the medications they have to receive. For this case, predefined SMSs will be stored in the HFP and they will be forwarded to the patients on specific time intervals, a few minutes prior of the exact time for receiving their medication, based on their patient profile, which is stored in the platform.
  Furthermore, a scheduler will also run at preset intervals, in order to detect delayed medical data entries from the sensors to the central database of the system. This function will be used in order to remind and notify the patients that the required readings from the sensors have not been received. This function will be based on patient profiles, where the scheduler will be checking whether a required sensor reading has been received on the desired time interval, specified by the superintendent doctor.

- Alerts: This function will also be used for transmitting warning SMSs in order to alert both the doctors and the patients, in case the readings received by the medical sensors used by a patient, show a recrudescence of their vital signs, or, in general, in case a serious health risk is detected by the system. These messages will be automatically triggered by the platform, if the data received from the sensors is below or above the desired levels, which are set individually for each patient. This kind of examination of the data is performed in the central database of the HFP. The SMS messages that are sent in this case are dynamically generated, providing the superintendent doctors with the option of writing specific SMS messages to their patients, including specific medical instructions.

- Notifications: Input "error" readings from the sensors will also trigger the messaging function when inaccurate data has been sent to the main database. For example, if a heartbeat reading exceeds the maximum possible level of heartbeats (e.g. 540 heartbeats, which is impossible for a human), then a predefined text message is transmitted, in order to provoke the patient to retransmit medical data from the sensors to the central database of the platform.

- Confirmations: If all the requested readings from the sensors have been successfully received, stored and properly examined by the system in the central database, a confirmation SMS is sent to the patient. The text message in this case is also predefined and the purpose of its transmission is basically to reassure the patient that all the readings have been properly received.

5.3 Functional specification and Integration of Decision Support Tools

Current open source technologies for supporting decisions generally handle information in form of rules.

A rule in natural language can be similar to "if the temperature of a patient is greater than 37 degrees then that patient has got the fever".

Of course a rule like the previous one can be used to ask to the system the list of patient that got the fever.

Now let's examine an extended case: suppose a patient is assuming a specific medicine and he is having a problem. Who/What is monitoring this new state and who/what carry on the following necessary steps? That is, who informs the system and/or the doctor about the actual state of the patient? And who/what tells the system it has to suggest the doctor some information?

The first step could be simply "alert the doctor that patient X has a problem".

Taking into account the Deliverable 5 we have to handle flows of information and procedures. Both static and dynamic information and procedures must be handled and also we have to keep in mind that flows can evolve.

Then, the HFP needs to include also a "controller" module, i.e. a module able to know, organize and assign the management of events.

A decision support system (DSS) able to apply a reasoning process to a specific question, connected to the KB and User levels, should interact also with the controller, i.e. with the middleware that should support the management of the workflow.
In this way, the functional specification of the middleware is therefore strictly related to the requirements of the DSS.
The controller that handles data and processes flows can be thought in different ways, as a workflow management or an ad-hoc system.

Given in input a formal description of *business* processes, a workflow management system takes care of maintaining the state of these processes execution by delegating the specific activities to persons and applications.
This helps the development of an interactive system able to support the solution of a problem and to improve the decisional process.

From the functional point of view, the workflow requirements can be distinguished in high-level, partially extracted by the descriptions of the HF scenarios, and low-level, related to the information flow and the different events to be managed. Thus, a workflow should:

- Organize the possible actions of the users
- Handle occurring events
- Execute device/computer processes
- Handle consequent actions from previous points using a system specific knowledge

Workflow systems able to design and manage series of tasks, states, events and actions have to be handled by the HF platform.

A Workflow defines a responsible for each task and sends/receives data from the responsible of the next task.

In the last decade workflows have been employed to automate business processes mainly thinking of applications. These systems didn’t properly consider interactions between human and processes. The need of modelling and monitoring in real time pushed to develop business process management systems.

A modern workflow should be able to provide the possibility to manage graphs of models, tasks, states, access, authorizations, monitoring, reporting.

Captured audit data can be used not only for statistic information but also to provide information used among processes for reporting monitored performances and by decision support systems able to deal and dynamically change on the base of these information.

New technologies such as XML, Web Services and Semantic Web opened new ways to workflows, according also to the Enterprise Service Bus vision (orchestration and choreography).

Some examples of products implementing Workflow Management Systems are listed in Figure 5.3.

Twister
jBPM
Enhydra Shark
OpenSymphony OSWorkflow
con:cern
Werkflow
ObjectWeb Bonita
Bossa
Open Business Engine
The Open for Business Workflow Engine
OpenWFE
WfMOpen
XFlow
JFolder
Taverna
Freefluo
Micro-Flow
JFlower
YAWL
Syrup
PXE
ActiveBPEL
Antflow
Dalma
Swish

Figure 5.3 - Java Open Source Workflow Management Systems

The advantage of this kind of systems is that they are independent of the flow and if it evolves during the time it's not necessary to rewrite the application that manages it, but only we need to redefine the flow using the process definition language supported by the workflow management system (the controller).
The process definition language allows to:

- map the activities and their relationships;
- define the criteria of start and end points of processes;
- define participants to the processes;
- define applications and data associated to processes.

It's true that this method requires a complex infrastructure but in case of complex flow it's use is worthwhile.

Another possibility would be to build an ad-hoc procedure that could monitor the information managed by the DSS: thus, after any information is inserted or updated in the knowledge base, this procedure should check sensible data in order to evaluate whether or not a successive action has to be performed.
For example, after data insertion regarding a patient, the monitoring application should evaluate the new scenario and eventually it should alert or suggest a doctor a modification of a treatment.

5.4 Bibliography and References

[1] G. Graefe: Query evaluation techniques on large databases – ACM Computing Surveys, 25(2):73-170, 1993

[2] S. Madden: Query Processing for Streaming Sensor Data, Univ. of California at Berkeley 2002

[3] J. Chen, D. DeWitt, F. Tian, Y. Wang: NiagarCQ: A scalable continuous query system for internet databases – ACM SIGMOD, 2000

[4] S. Madden, M. A. Shah, J. M. Hellerstein, W. Hong: Continuously adaptive continuous queries over data streams – ACM SIGMOD, Madison, WI, June 2002

[5] P. Seshadri, M. Livny, R. Ramakrishnan: The design and implementation of a sequence database system – VLDB, 1996

[6] Observations and Measurements – Open GIS Consortium Discussion Paper v. 0.9.2, 2003

[7] Eyes Project Deliverale 1.1: System Architecture Specification – October 2002

6. Modelling of the Middleware

6.1 Design View

6.1.1 Design view of the HF communication infrastructure

As already anticipated in Section 4.3, the major components taking part in functional schema of the HEARTFAID platform could be identified in three macro systems:

- Healthcare Information System (HIS)
- Remote monitoring infrastructure, which will adhere to the Ambient Intelligent (AmI) paradigms and principles
- Decision Support Systems (DSS) and Knowledge Discovery in Databases (KDD) Systems

Notice that while it can be identified well in which environment the first two systems have to operate (medical and patient environment), and the KDD systems also (medical and technological research environment), as far as the DSS system is concerned, its support is cross-sectional on all environments. For example a DSS can support a Cardiologist in discover points of interest on images or signals, or also, a DSS integrated in the AmI platform can forecast in real-time monitoring a critical condition of a patient. We make this explanation because in the next schemas we represent DSS/KDD systems as architectural isle that has merely a logical separation.

On the other hand, this three part subdivision of the HEARTFAID platform was established to maintain the logical separation between systems, separation that guarantees flexibility and extensibility of the entire architecture. In fact, the HEARTFAID platform could be composed by an indeterminate number of HISs commercially or freely available, or an indeterminate number of Cardiology EPR software modules connected on the bus of HIS (moving toward the EHR vision also), or different monitoring platforms. The role of the HEARTFAID middleware is to build the infrastructure making easily integrable these different systems both at data level and end-user level, and also both at macro-system level (by way of MOMs) and at HEARTFAID platform level (by way of Service Bus).

According to this HEARTFAID vision, in the following figures it is shown a typical HIS and an AmI platform respectively, both based on a bus topology (see Section 3). Of course is not objective of the HF project to implement a HIS; on the contrary, the project has the main goal to design an EHR to be implemented in the cardiovascular contexts addressed by the project itself. Much effort will also be spent for the study, design and implementation of the AmI component, as well as of the component responsible for the knowledge management.

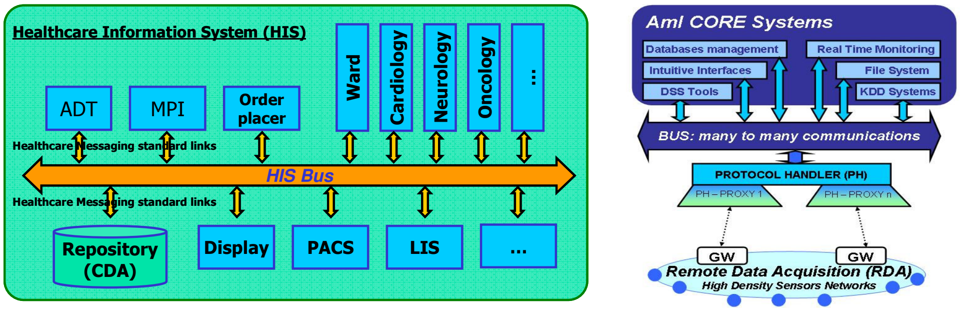

Figure 6.1 - On the left side a typical HIS architecture and on the right side a typical AmI architecture.

These architectural schemas will be the reference we will follow in HEARTFAID middleware(s) implementation.

For each system, every software module is connected to the appropriate bus and exchange messages. As example in HIS the Admission Discharge Transfer (ADT) IHE actor (see Sections 3 and 4.6) is implemented by a software module connected to the HIS Bus that sets up the HIS integration middleware. When a patient has been admitted, ADT publishes on bus an A01 HL7 message, and all subscribed actors will receive the message. We can consider the buses as a message oriented middleware that takes part in each macro-system.

On the other hand, to satisfy requirements of modularity, composability, componentization, and interoperability, the HEARTFAID platform will be a service-oriented platform (see Section 3.3). In this approach every macro-system will expose a set of services on the HEARTFAID service bus, as showed in Figure 6.2.

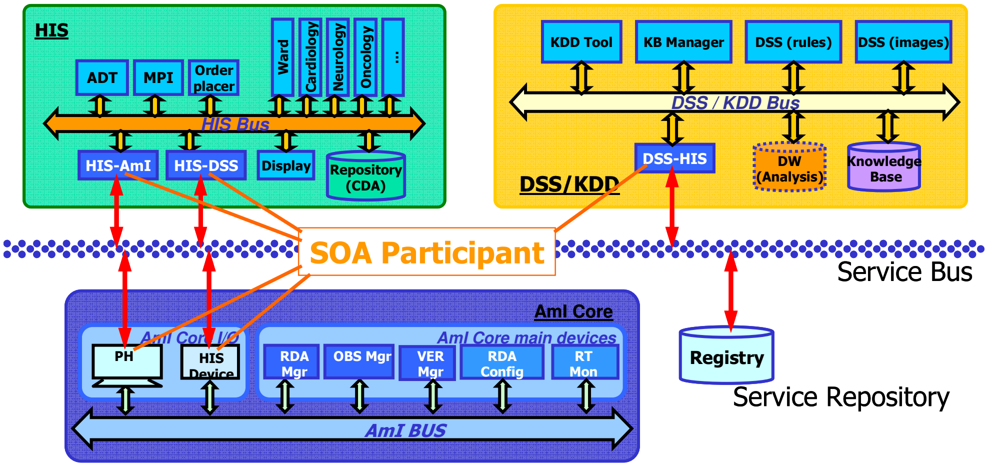

Figure 6.2 - A schematic view of the entire HEARTFAID platform

This service oriented architecture is designed to integrate macro-systems by way of software modules that on one side are connected to their bus, and on the other side expose services on the HEARTFAID **service bus**.

Synapsis, together with other EU partners, is already experimenting the AmI concepts in the agro-food field in the context of the Integrated project GOODFOOD (FP6-IST-1-508774-IP).
The GOODFOOD project addresses the issues of integrating wireless sensors networks, implemented with the Sensor Web technology, wireless communications and a software framework that provides distributed databases, intelligent and intuitive systems and interfaces, to support the assessment of healthiness and quality of food as well as the information spreading. The solution that is being developed comprises a Remote Data Acquisition (RDA) for gathering information over a sensed environment, a communication infrastructure transporting data across the actors of the framework and a software component (AmI Core) represented by a set of systems involved in storage, monitoring, intelligent analysis and presentation of the data. In particular, the AmI Core is based on a scalable and
open many-to-many communication scheme, called AmI Bus and implemented using a bus-like topology. Entities, called AmI Devices, involved in implementing specific functionalities (i.e. storing, analyzing, monitoring and presenting data), are able to automatically connect to the AmI Bus and exchange information by using a messaging mechanism. A particular AmI Device called Protocol Handler (PH), is responsible for bridging the AmI Core to the RDA and vice versa.

Of course, the approach that being experimented in the framework of GOODFOOD project, will need to be re-engineered in order to be specifically tailored and configured on the specific requirements of the HEARFAID project, as well as finely tuned for the remote monitoring of patients with heart failure.

It is worth it to highlight that the use of a Service BUS has a threefold advantage:

1) It allows to integrate different data related to the same patient;
2) It allows managing the information related to patients separately from the management of the domain knowledge;
3) The module responsible to provide decision support is able to integrate the real data with the knowledge encoded in the HFP.

This way, there will be components, either internal or external to the platform, such as for example the HIS or the AmI infrastructure for the acquisition of measurements, that will feed the HFP and, consequently, also the knowledge management system, and components, such as the DSS, that using the middleware of the HFP will be able to integrate the encoded knowledge with the data of the patient in order to provide decision support to the clinicians and, in the end, a better care to the patients.

To demonstrate the feasibility of this architectural design, we made some preliminary implementations. The integration was made between the HIS and the AmI platform, and the following aspects have been taken into account:

- New patient enrolment
- New monitoring data available

The involved software modules implemented in this early experimental integration are:

- on AmI platform side:
  - Protocol Handler (PH): it handles (by way of proxies) the communication protocol with other systems (see Figure 6.3), such as the Remote Data Acquisition (RDA) networks, or other software systems (e.g. the HIS);
  - HIS Device: it publishes an interface to get observations provided by RDA and stored by the Observation Manager device
- on HIS side:
  - HIS-AmI-Config: it allows the re-configuration of the RDA entities, and can be considered as the gateway (GW) of HIS (GWN in Figure 6.3)
  - HIS-AmI-Monitor: it allows getting the data measures collected by RDA

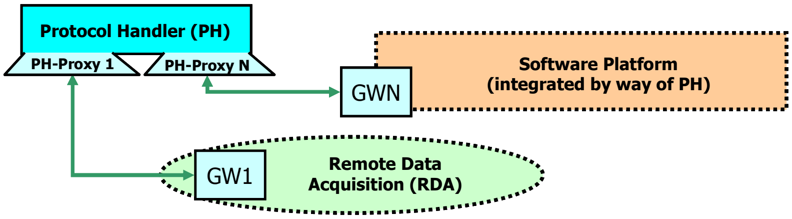

Figure 6.3 - The Protocol Handler (PH) can handle the communication with gateways of RDA and of other software platforms

A patient can be seen as an entity and we modelled the patient enrolment as a new message delivered to the correct Proxy of PH (see Figure 6.4). The Electronic Patient Record (EPR) has to enrol a new patient for home monitoring, and ask to Master Patient Index (MPI) actor the ID of the patient. The HIS-AmI-Config notifies to PH of AmI platform there is a new entity in Remote Data Acquisition with this ID. Or the HIS-AmI-Config notifies to PH there is a new sensor on patient therefore there is a new entity in RDA.

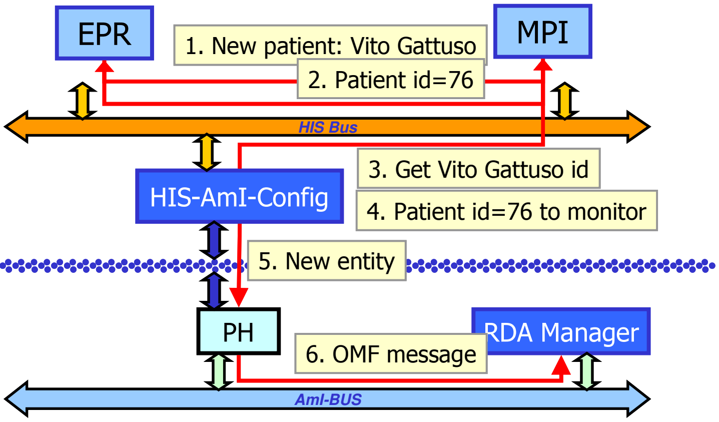

Figure 6.4 - New patient enrolment flow in integrated systems

Moreover, when patient's entities are correctly configured in the RDA, on the basis of configuration, new information about measures are delivered to HIS. The HIS Device receives new observations from Observation Manager by way of AmI Bus and it sends new observations to EPR on scheduled time, or on demand by HIS (by way of the module HIS-AmI-Monitor). Finally the HIS-AmI-Monitor stores in Electronic Patient Record (EPR) new measurements available (see Figure 6.5).

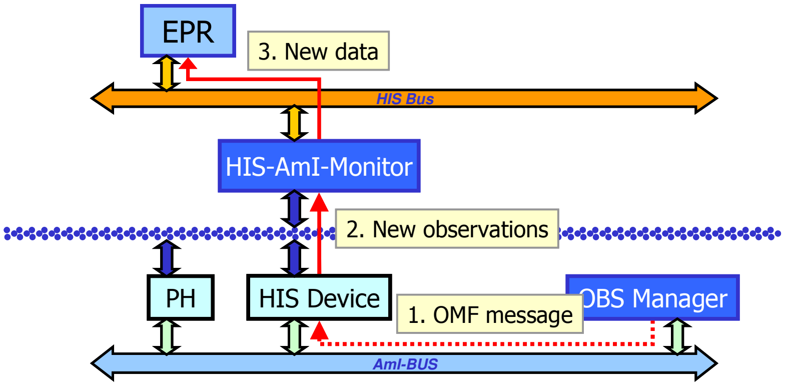

Figure 6.5 - New observation available flow

6.1.2 Design view of the HF Remote Data Acquisition networks

The design elements which comprise the HR Remote Data Acquisition network can be decomposed as follows:

1. The sensors/medical devices, which perform the measurements and produce the biomedical data.
2. The communication channels that the sensors use in order to transmit the biomedical data to another end point. The communication can be wireless or wired, according to the specifications supported by each sensor.

3. The data acquisition device, which is responsible to collect the data from the sensors in its neighbourhood. This device can be a PC in the user's home, a PDA or a mobile phone. The data acquisition device is responsible, not only to collect the data, but also to enhance and transmit the data to the HEARTFAID platform. For this reason, the middleware functionality described in Section 5.2 has to be implemented in the data collection device.

When examining possible data acquisition scenarios we can focus on the following two cases:

a) The InHome scenario, according to which, a PC will be responsible to gather the biomedical data from the sensors, enhance it and finally send it to HEARTFAID platform in the form of XML messages. Consequently, in this scenario all the middleware functionality can be executed by the PC.

   The interaction between the InHome PC and the central middleware server (as it is described in section 6.1) is portrayed in detail, in the following activity diagram.

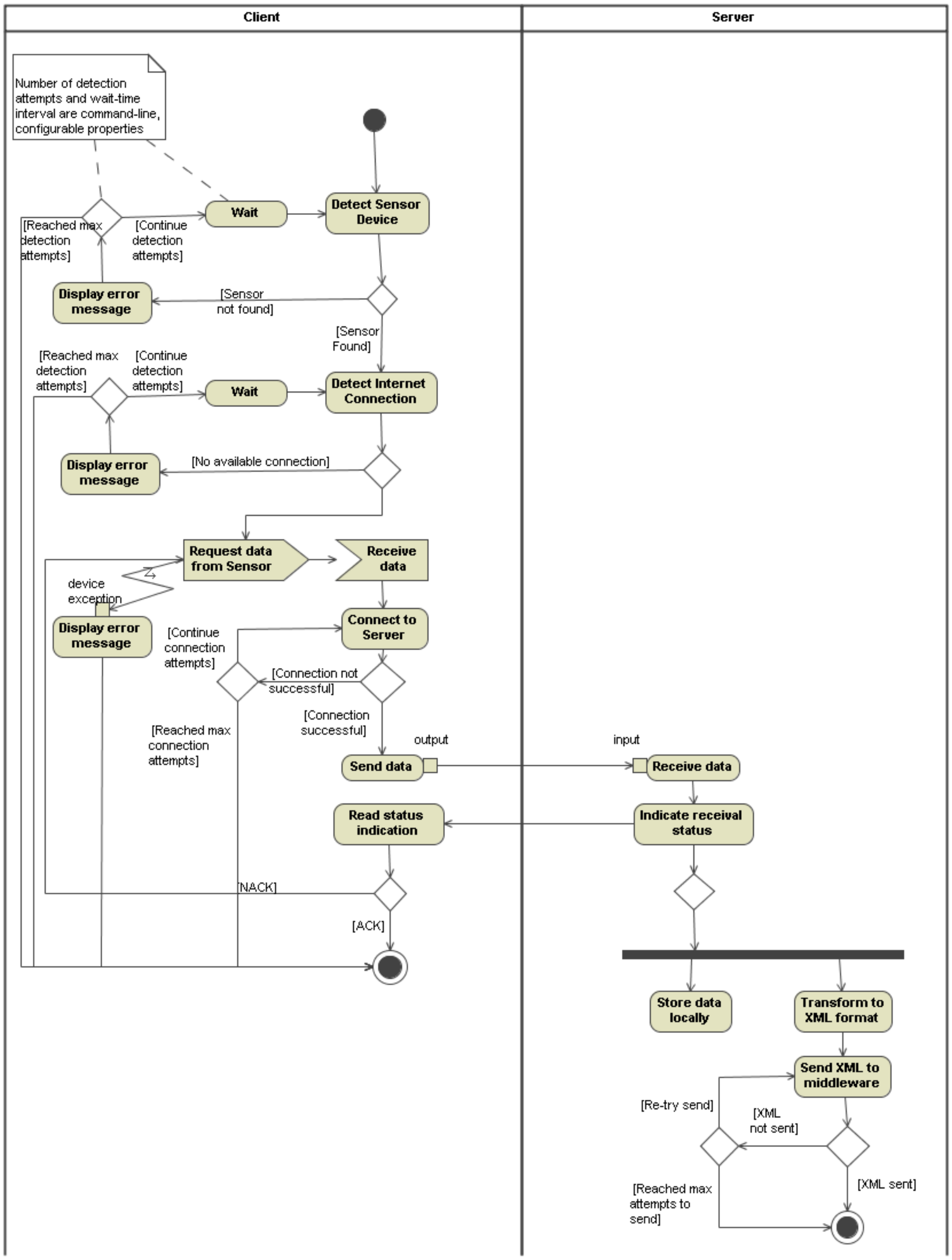

Figure 6.6 - Activity diagram of the interaction between the InHome PC and the central middleware server

b) The On-the-move scenario, where the data acquisition device has to operate in an outdoor environment. Undoubtedly, state of the art smart-phones provide all the processing power, storage capabilities and communication features that enable them to collect, enhance and transmit the data in the same way as a PC at home. For example, a mobile phone executing J2ME applications and also supporting the JSR-82 (bluetooth connectivity) and JSR-75 (PIM and File Connection API) can gather data from sensors via bluetooth, locally store the data and transmit it in XML messages using well known Java XML APIs (e.g. KXML) over IP using GPRS/3G networks. However, it was preferred to limit the software functionality executed on the phone only to the collection of data and execute the remaining operations in an intermediate server, which receives the collected data over IP, enhances it and sends the corresponding XML messages to the HEARTFAID platform. The main reason for following such an approach is to reduce the amount of data transmitted by the mobile phone to only the necessary values instead of transmitting large XML messages directly to the HEARTFAID platform.

The block diagram of the Remote Data Acquisition Network is depicted in Figure 6.7.

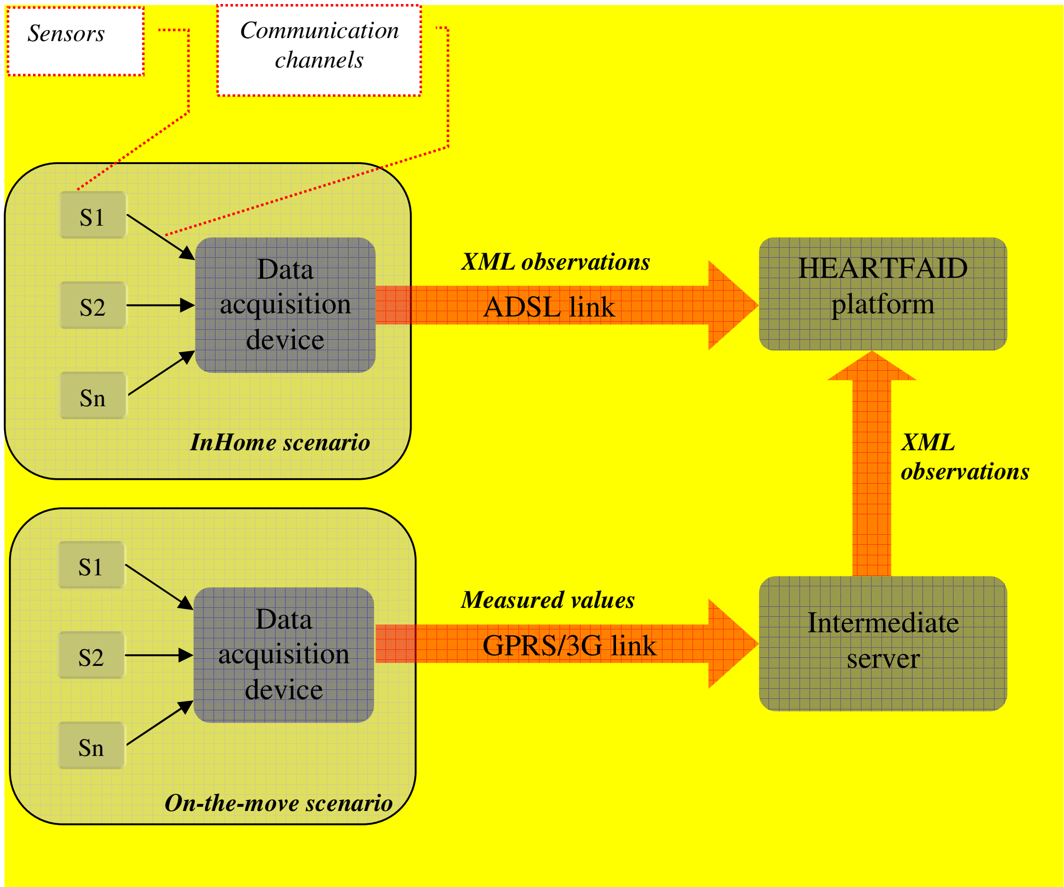

Figure 6.7 - Block diagram of the Remote Data Acquisition Network.

6.1.3 Design view of the HF end-user applications

The interaction of the end users (clinicians and patients) with the main components/functions of the central middleware platform can be depicted as in Figure 6.8 below:

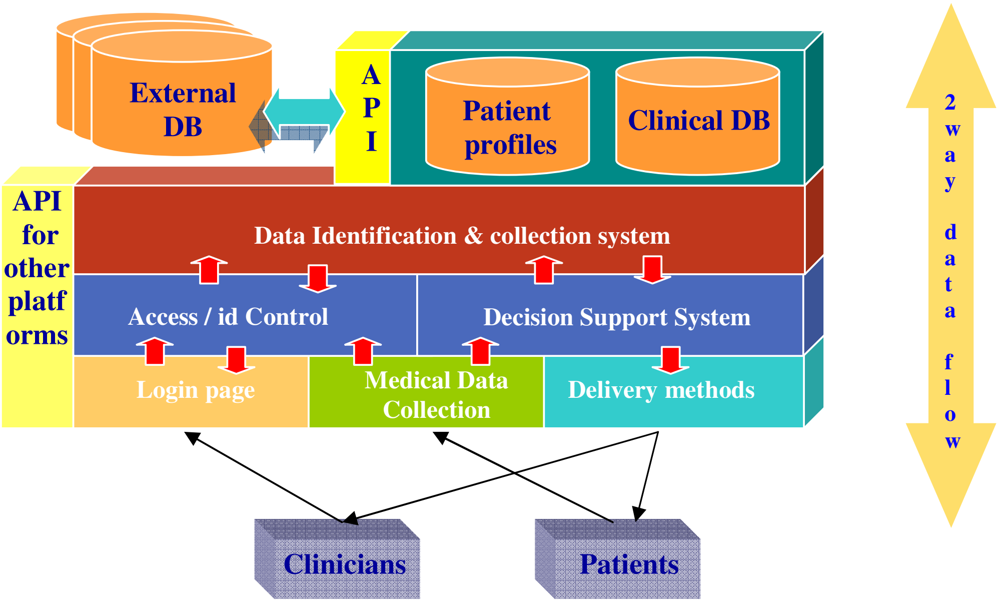

Figure 6.8 - Interaction of the end users with the main components/functions of the central middleware

Alert and alarm system

Design view of one way and two way communications, regarding SMS messages transmitted to mobile users (as described in Section 5.2), are portrayed in the following two Figures:

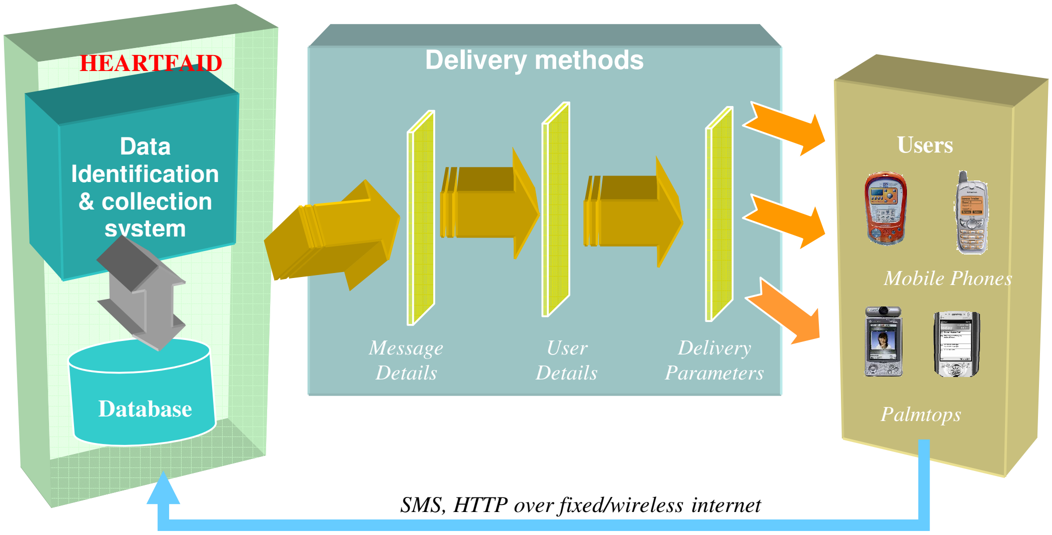

Figure 6.9 - One way SMS communication

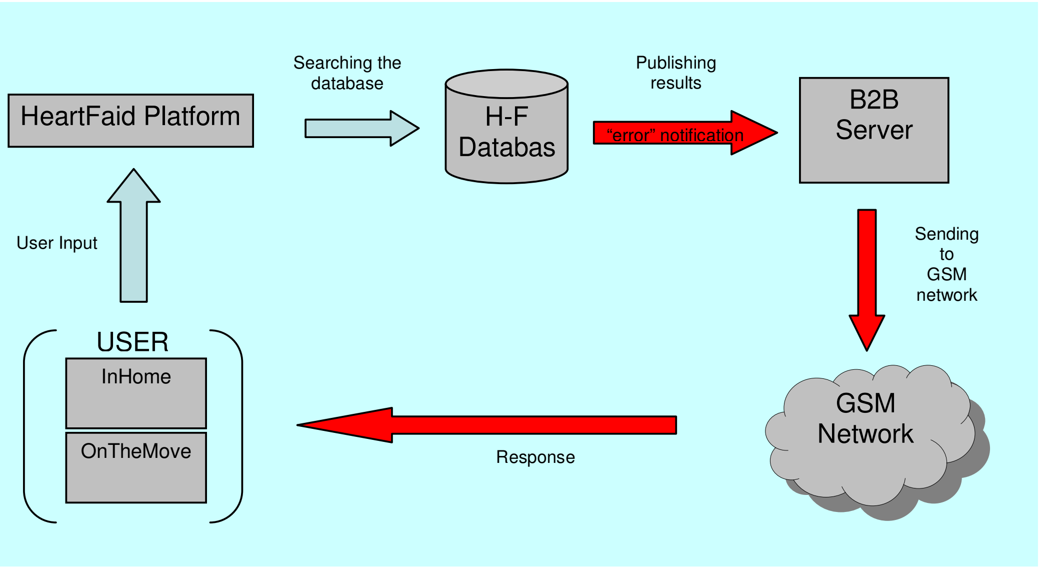

Figure 6.10 - Two way SMS communication

6.2 Hardware Specifications

In the case of the on-the-move scenario, where the data is collected by a mobile phone, the data can be manipulated as follows according to the data volume and nature:

a) **Case of large data volume (e.g. continuous mesurements over a large period of time)**

In case of continuous measurements involving large volume of data, the mechanisms that could be applied are:

i. The mobile phone could act as a data logger to store the whole data and transmit it to the HEARFAID platform via the ADSL link when the user returns to home. This scenario requires large storage capacity on the mobile phone and could be made feasible in the case where the mobile phone supports usage of additional memory in the form of a memory card and also J2ME is allowed to access the storage repository of the mobile phone (i.e. the phone supports the JSR-75 PIM and FileConnection)

ii. The mobile phone can transmit the data directly to the HEARTFAID platform over a GPRS/3G connection (through an intermediate server). This scenario requires a reliable connection between the mobile phone and the intermadiate server. Of course, incoming data can be temporarily stored in the mobile phones Record Management System (RMS) to cope with temporary loss of connectivity.

b) Case of small data volume (single measurements)

In the case of small volume of incoming data (for example single mesurements), the mobile phone can directly transfer the measured data to the HEARTFAID platform via the GPRS/3G connection, provided that the connectivity exists. In case that there is no connectivity at the time of measurement, the measured value(s) can be temporarily stored on the mobile phone ant transferred to the HEARTFAID platform as soon as the connectivity exists.

7. Security Issues

7.1 Safety, Security and Privacy in Healthcare

The notion of **Safety** is usually associated with the prevention of the harm (or damage) that may be done to human life (or an environment) by a system. In the context of computer-based systems safety is defined as the freedom from unacceptable risk of harm. In this definition, the terms harm and risk have particular interpretations, and these are:

- harm is the physical injury, or the damage to health or property, that may be caused by the system;
- risk is the probable rate of occurrence of a hazard (a hazard is understood to be a situation in which there is a potential for human injury) causing harm and the degree of severity of the harm.

Thus, the concept of risk has two elements:

- the frequency with which a hazard occurs;
- the consequences of the hazardous event.

Security is the protection of information systems against unauthorized access to or modification of information, whether in storage, processing, or transit, and against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.

The term security in the information technology domain is commonly understood to mean the protection of a system's assets as the information and the information processing resources. Specifically, security has been defined as the preservation of the confidentiality and integrity of the information systems and the data that they maintain as well as ensuring the accountability and availability of the services and data.

A basic element of the security in healthcare is the health information which is defined as any information, whether oral or reported in any form or medium, that is created or received by a health care provider, public health authority, life insurer and relates to:

- the past, present, or future physical or mental health or condition of an individual;
- the provision of health care to an individual;
- the past, present, or future payment for the provision of healthcare to an individual.

Due to the sensitivity of the information maintained by the component information systems, advanced security and authentication methods need to be employed.

The notions of safety and security may appear to be different, but they are similar in the sense that:

- they concerned with freedom from different types of undesirable incidents. Safety is predominantly concerned with preventing the occurrence of harm to human life, and security prevents unauthorised access to information;

- both define a concept of a risk. The risks to a system are either eliminated or made as low as reasonable practicable.

Privacy, on the other hand, includes the right of individuals and organizations to determine for themselves when, how and to what extent information about them is communicated to others.

Individually identifiable health information is any health information that:

- identifies the individual, or
- there is a reasonable basis to believe that the information can be used to identify the individual.

At one end of the spectrum of identifiability, data are completely anonymous and not linked to any identifiers. This is the least sensitive type of information. However, depending on how the anonymization process is carried out, some risk may remain for the anonymized data to be re-identified (e.g. through processes such as data matching).

Next in the spectrum, identifying data are transformed into pseudo-identifiers, i.e. identifiers that cannot be linked to any real person or organization. In terms of privacy, this type of data may be viewed as equivalent, as far as sensitivity is concerned, to completely anonymous data, depending on the cryptographic techniques being used.

Further down the spectrum, is the code-linked information where the identifiers are replaced with a code that, whenever necessary, can be linked to information that reveals an individuals' identity. From a privacy standpoint, and depending on how the codes used to re-identify the data are created (e.g. with simple conversion tables) and controlled, this information is more sensitive than completely anonymous data or than de-identified data linked to pseudo-identities.

Finally, at the opposite end of the spectrum is the completely identifiable information. From a privacy perspective, this is the most sensitive type of information, which is associated to the greatest risk.

A couple of basic approaches to safeguarding confidentiality have been identified in the past. The first approach focuses on the creators and maintainers of the information, prohibiting them from disclosing the information to inappropriate parties. An alternative approach focuses on the use of so called Privacy-Enhancing-Techniques (PETs) and other measures using cryptographic techniques. In contrast with horizontal types of data exchange (e.g. for direct care), vertical communication scenarios (e.g. in the context of disease management studies and other research) do not require identities as such: here the use of pseudo-IDs can help find solutions.

Privacy enhancing solutions range from very simple to complex technical and nontechnical methods and measures.

Examples of these techniques are (non exhaustive list):

- Hard de-identification at the source side by the owner of the data;
- Various types of anonymisation and/or pseudonymisation

- De-identification may be reversible or irreversible, conducted with or without the help of a Trusted Third Party, in batch or in interactive mode, etc.;
- Controlled database dilution/perturbation, which consists in injecting fake data in a controlled way;
- Data (flow) segmentation;

7.2 Security Aspects

Confidentiality. Patient confidentiality is breached whenever anyone other than the patient, the patient's physician or other healthcare professional directly responsible for the patient's care learns any private patient information. An IT application must preserve the confidentiality of the patient medical data by the development and the deployment of advanced security mechanisms.

Integrity. Data integrity is defined as the property that data has not been altered or destroyed in an unauthorized manner.

In order for patients to be treated appropriately for their medical condition, physicians and other clinical staff must have accurate and complete information on the patient. Healthcare professional rely upon and trust the information stored in the Electronic Healthcare Record systems or transfer through telemedicine applications.

There are also other applications where integrity will need to be assured, such as when transmitting an email message, examination orders and results, healthcare record segments, etc. between two parties.

Availability. IT applications and their stored data must be always available and useable upon demand by an authorized user or system.

Accountability. IT applications for healthcare and their users, i.e. the healthcare professional, must be accountable for their actions concerning the treatment of a patient. This means that someone has to be always able to trace the actions of the healthcare professionals that are related to the usage of the IT application (i.e. maintaining audit logs). This is important not only for the data insertion or modification to the Electronic Healthcare Record systems, but also for the data that are shared during a telemedicine session including the opinions that are exchanged.

Accountability and confidentiality are closely related to authorization that refers to the process of identifying a user and granting privileges to the use and his processes. Once a user is identified (reliably), the privileges, rights, property, and permissible actions of the user are determined by authorization.

7.3 Threats and vulnerabilities

A **threat** is an indication of a potential undesirable event. It refers to a situation in which a person could do something undesirable (a hacker initiating a denial-of-service attack against a hospital's email server) or a natural occurrence could cause an undesirable outcome (a fire damaging a hospital's information technology hardware). Threats consist of the following properties:

- Asset – something of value to the enterprise that is in danger by a threat.

- Actor – who or what may violate the security requirements (confidentiality, integrity, availability) of an asset.
- Motive (or objective) – defines whether the actor's intentions are deliberate or accidental.
- Access – how an actor will access the assets (network access, physical access).
- Outcome – the immediate outcome (disclosure, modification, destruction, loss, interruption) of violating the security requirements of an asset.

In order to access an information asset to affect a desired outcome, a threat must take advantage of vulnerability. Vulnerability is a weakness in an information system, system security practices and procedures, administrative controls, internal controls, implementation, or physical layout that could be exploited by a threat to gain unauthorized access to information or disrupt processing.

Technology vulnerabilities are present in and apply to network services, architecture, operating systems, and applications. Technology vulnerabilities are often grouped into three categories:

- *Design vulnerabilities* – a vulnerability inherent the design or specification of hardware or software whereby even a perfect implementation will result in vulnerability.
- *Implementation vulnerabilities* – a vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design.
- *Configuration vulnerabilities* – a vulnerability resulting from an error in the configuration and administration of a system or component.

In addition, the technology vulnerability evaluation identifies technical vulnerabilities that can be used to refine the picture of organizational vulnerabilities. An organizational vulnerability can be attributed to the lack of well defined procedures at the healthcare organizations or from a user's lack of awareness regarding information security policy and practice, deliberate avoidance or circumventing of existing policy and practice, insufficient training and readiness to address information security vulnerabilities, misplaced or inappropriate trust, etc.

7.4 Communication Security

For communication between different information domains, a trusted end-to-end communication policy must be established. In general, access rights can be managed through:

- authentication, being the process of ensuring that the communicating party is the one claimed to be;
- authorization, being the process of ensuring that the communicating party is eligible to request for a specific action.

In addition audit trails are needed to ensure accountability of actions of individual persons or entities, such as obtaining informed consent or breaching confidentiality.

These records can be used to reconstruct, review, and examine transactions, track system usage, control authorized users, detect and identify intruders.

The ISO Technical Committee 215 (ISO/ TC215) on its Technical Report 21089 (ISO/ TC 215, 2004) offers a guide to trusted end-to-end information flow for health (care) records and to the key trace points and audit events in the electronic entity/ act record lifecycle (from point of record origination to each ultimate point of record access/ use). It also offers recommendations regarding the trace/ audit detail relevant to each.

ENV 13608 (CEN/ TC 251 ENV 13608-1:2000) (CEN/ TC 251 ENV 13608-2:1999) on Security for healthcare communication specifies a methodology for defining, expressing and selecting a communication protection profile specification (i.e. integrity, confidentiality, availability, and legal accountability), defines a standard way of securing healthcare objects (so that they can be transported over open, unsecured networks, or stored in open unsecured repositories), and specifies services and methods for securing interactive communications used within healthcare (including preservation of data integrity, confidentiality with respect to the data being communicated, and accountability in terms of authentication of one or both communicating parties).

7.5 Authentication, Authorization and Digital Signature

Currently the most common technological tool to cover various security aspects is the Public Key Infrastructure (PKI). PKI is used to describe the processes, policies, and standards that govern the issuance, maintenance, and revocation of the certificates, public, and private keys the encryption and signing operations require. PKI incorporates the necessary techniques to enable two entities that do not know each other to exchange information using an insecure network such as the Internet.

PKI is based upon asymmetric cryptography and each entity (user, information system, etc.) is provided with a pair of keys (a private and public one).

The public key security infrastructure comprises the following services:

- Certification Authorities that control and manage the PKI, publish public key certificates, and impose policies in their domain of authority;
- Registration Authorities that act on behalf of the Certification Authorities in order to declare registered in the domain of authority the Certification Authority manages;
- Certificate Management Systems for management of certificates during their entire duration of validity;
- X.500 directories that store public key certificates and public information for the holders of certificates, and are used for the verification of digital certificates;
- User Certificate for each of the users, which is published by the Certification Authority, and is stored together with the user’s private key, in a microprocessor card.

In order to guarantee the authenticity of a set of input data, the same way a written signature verifies the authenticity of a paper document, PKI uses digital signatures.

European Prestandard (ENV) 13729 (CEN/TC251, 1999) on Secure User Identification for Healthcare Strong Authentication Using Microprocessor Cards defines how certificates are used in order to support authentication. Because of its importance ENV 13729 is expected to be reviewed and enhanced further in the future.

ISO/ TC 215 is going to create a new standard (22600) on Privilege Management and Access Control (ISO/ TC 215, Health Informatics, Privilege Management and Access Control, 2005) including structural and functional roles (e.g. delegation policies) which is very important for accessing a complex multi-lingual and multimedial virtual distributed EHR system.

Building on the digital signature technology, the digital signing of clinical documents is a special instance where the nature of the clinical workflow may require that each participant only signs that portion of the document for which he/she is responsible. Older standards for digital signatures do not provide the syntax for capturing this sort of high-granularity signature or mechanisms for expressing which portion a party wishes to sign.

7.6 Transmission of medical data

Regarding the transmission of medical data from the sensors, which are placed at the patient's home, to the central middleware (database) of the platform, there are various steps which require different kinds of security protocols.

In the case of medical data acquisition from the sensors to the "client pc", where the raw data from the sensors is going to be gathered and is based in the patient's home, there are two basic ways for data transmission, according to the specified sensors that are going to be used:

- Bluetooth sensors
- Serial port sensors (RS-232 port)

7.6.1 Bluetooth security:

In every Bluetooth device, there are four entities used for maintaining the security at the link level, between the sensor and the client pc. These are:

1. The Bluetooth device address (BD_ADDR), which is a 48-bit address that is unique for each Bluetooth device and it was defined by the Institute of Electrical and Electronics Engineers (IEEE).
2. Private authentication key, which is a 128-bit random number used for authentication purposes.
3. Private encryption key, 8-128 bits in length that is used for encryption, and
4. A random number (RAND), which is a frequently changing 128-bit random or pseudo-random number that is made by the Bluetooth device itself.

In Bluetooth Generic Access Profile, the Bluetooth security is divided into three modes:

- Security Mode 1: non-secure
- Security Mode 2: service level enforced security

- Security Mode 3: link level enforced security

The difference between Security Mode 2 and Security Mode 3 is that in Security Mode 3 the Bluetooth device initiates security procedures before the channel is established.

There are also different security levels for devices and services. For devices, there are 2 levels, "trusted device" and "non-trusted device". The trusted device obviously has unrestricted access to all services. For services, 3 security levels are defined: services that require authorization and authentication, services that require authentication only and services that are open to all devices. For the link between the medical sensors and the client pc we will consider only the first 2 security levels mentioned above.

Key management

All security transactions between two or more parties are handled by the link key. The link key is a 128-bit random number. It is used in the authentication process and as a parameter when deriving the encryption key. The lifetime of a link key depends on whether it is a semi-permanent or a temporary key. A semi-permanent key can be used after the current session is over to authenticate Bluetooth units that share it. A temporary key lasts only until the current session is terminated and it cannot be reused. Temporary keys are commonly used in point-to-multipoint connections, where the same information is transmitted to several recipients.

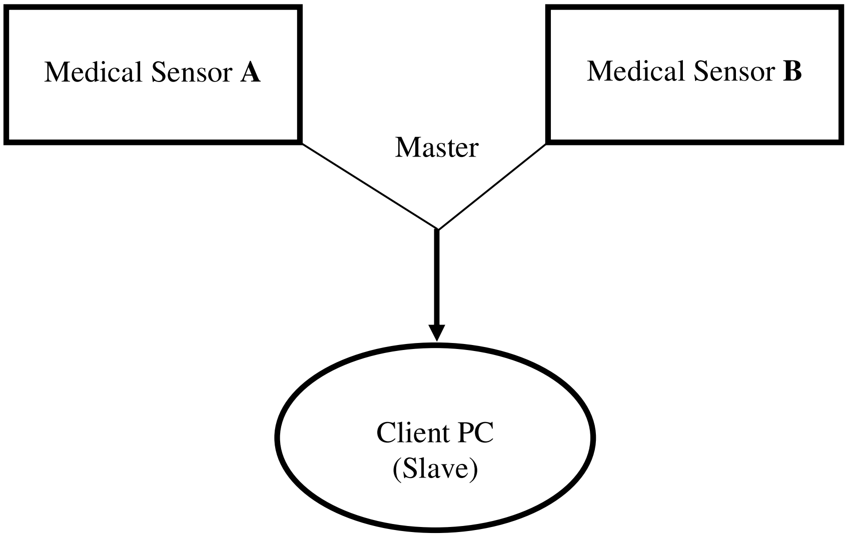

Figure 7.1 - Link keys between devices

There are several different types of keys defined in Bluetooth. Link keys can be combination keys, unit keys, master keys or initialization keys, depending on the type of application. In addition to link keys, there is the encryption key.

The unit key is generated in a single device when it is installed. The combination key is derived from information from two devices and it is generated for each new pair of Bluetooth devices. The master key is a temporary key, which replaces the current link key. It can be used when the master unit wants to transmit information to more than one recipient. The initialization key is used as link key during the initialization process when there are not yet any unit or combination keys. It is used only during the installation.

The length of the Personal Identification Number (PIN) code used in Bluetooth devices can vary between 1 and 16 octets. The regular 4-digit code is sufficient for some applications, but higher security applications may need longer codes. The PIN code of the device can be fixed, so that it needs to be entered only to the device wishing to connect. Another possibility is that the PIN code must be entered to the both devices during the initialization.

The initialization key is needed when two devices with no prior engagements need to communicate. During the initialization process, the PIN code is entered to both devices. The initialization key itself is generated by the E22 algorithm, which uses the PIN code, the Bluetooth Device Address of the claimant device and a 128-bit random number generated by the verifier device as inputs. The resulting 128-bit initialization key is used for key exchange during the generation of a link key. After the key exchange, the initialization key is discarded.

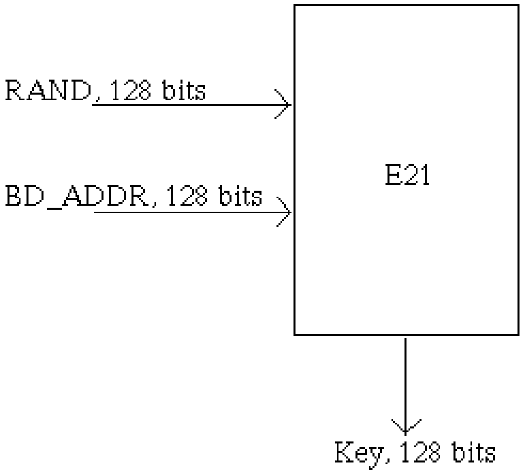

Figure 7.2 - Key generating algorithm E21 for unit and combination keys

The unit key is generated with the key generating algorithm E21 when the Bluetooth device is in operation for the first time. After it has been created, it will be stored in the non-volatile memory of the device and is rarely changed. During the initialization process, the application decides which party (medical sensors – client pc) should provide its unit key as the link key.

The combination key is generated during the initialization process if the devices have decided to use one. It is generated by both devices at the same time. First, both of the units generate a random number. With the key generating algorithm E21, both devices generate a key, combining the random number and their

Bluetooth device addresses. After that, the medical sensor and the client pc exchange securely their random numbers and calculate the combination key to be used between them.
The master key is the only temporary key of the link keys described above. It is generated by the medical sensor device, by using the key generating algorithm E22 with two 128-bit random numbers. As all the link keys are 128 bits in length, the output of the E22 algorithm is also 128 bits. The reason for using the key generating algorithm in the first place is just to make sure the resulting random number is random enough. A third random number is then transmitted to the client pc (slave) and with the key generating algorithm and the current link key an overlay is computed by both the master and the slave. The new link key (the sensor's key) is then sent to the client pc, bitwise XORed with the overlay. With this, the client pc can calculate the master key.

The encryption key is generated from the current link key, a 96-bit Ciphering Offset Number (COF) and a 128-bit random number. The COF is based on the Authenticated Ciphering Offset (ACO), which is generated during the authentication process. When the Link Manager (LM) activates the encryption, the encryption key is generated. It is automatically changed every time the Bluetooth device enters the encryption mode.

Encryption
The Bluetooth encryption system encrypts the payloads of the packets. This is done with a stream cipher E0, which is re-synchronized for every payload. The E0 stream cipher consists of the payload key generator, the key stream generator and the encryption/decryption part.
Depending on whether a device uses a semi-permanent link key or a master key, there are several encryption modes available. If a unit key or a combination key is used, broadcast traffic is not encrypted. Individually addressed traffic can be either encrypted or not. If a master key is used, there are three possible modes. In encryption mode 1, nothing is encrypted. In encryption mode 2, broadcast traffic is not encrypted, but the individually addressed traffic is encrypted with the master key. And in encryption mode 3, all traffic is encrypted with the master key.
As the encryption key size varies from 8 bits to 128 bits, the size of the encryption key used between the two devices must be negotiated. In each device, there is a parameter defining the maximum allowed key length. In the key size negotiation, the medical sensor device sends its suggestion for the encryption key size to the client pc. The client pc (slave) can either accept and acknowledge it, or send another suggestion. This is continued, until a consensus is reached or one of the devices aborts the negotiation. The abortion of the negotiation is done by the used application. In every application, there is defined a minimum acceptable key size, and if the requirement is not met by either of the participants, the application aborts the negotiation and the encryption cannot be used. This is necessary to avoid the situation where a malicious device forces the encryption to be low in order to do some harm.

Authentication

The Bluetooth authentication scheme uses a challenge-response strategy, where a 2-move protocol is used to check whether the other party knows the secret key. The protocol uses symmetric keys, so a successful authentication is based on the fact that both participants share the same key. As a side product, the Authenticated Ciphering Offset (ACO) is computed and stored in both devices and is used for cipher key generation later on.

First, the verifier sends the claimant a random number to be authenticated. Then, both participants (medical sensor device and client pc) use the authentication function E1 with the random number, the claimants Bluetooth Device Address and the current link key to get a response. The claimant sends the response to the verifier, who then makes sure the responses match.

If the authentication fails, there is a period of time that must pass until a new attempt of authentication can be made. The time period doubles for each subsequent failed attempt from the same address, until the maximum waiting time is reached. The waiting time decreases exponentially to a minimum when no failed authentication attempts are made during a time period.

7.6.2 Serial Port security

Serial communication is a popular means of transmitting data between a computer and a medical sensor. Serial communication uses a transmitter to send data, one bit at a time, over a single communication line to a receiver. Serial communication is popular because most computers have one or more serial ports, so no extra hardware is needed other than a cable to connect the medical sensor device to the client pc. In addition, the distance between the sensor device and the client pc will be very small, since both will be located in the patient's home. From the above we can understand that the security level in this case of medical data transmission is very high.

7.7 Bibliography and References

[1] Bluetooth Security by Christian Gehrmann (Author), Joakim Persson (Author), Ben Smeets (Author) Publisher: Artech House Publishers; 1 edition (2004)

[2] Wireless Security by Merritt Maxim, David Pollino. Publisher: Osborne/McGraw-Hill; 1st edition (April 29, 2002)

[3] http://www.niksula.hut.fi

[4] http://www.securityfocus.com

[5] http://www.bluetooth.com

[6] www.codebluecommunications.com

[7] www.infosecwriters.com

[8] CEN TC 251, NC 98-106: "First Working Document of Safety and security related software quality standards for healthcare", 1998.

[9] CEN TC 251, ENV 12924: "Medical Informatics - Security Categorisation and Protection for Healthcare Information", 1997.

[10] CEN TC 251, ENV 13608-1:2000, Health informatics – Security for healthcare communication – Part 1: Concepts and terminology, 2000.

[11] CEN TC 251, ENV 13608-2:1999, Health informatics – Security for healthcare communication – Part 2: Secure data objects, 1999.
[12] CEN TC 251, ENV 13608-3:1999, Health informatics – Security for healthcare communication – Part 3: Secure data channels, 1999.
[13] CEN TC 251, ENV 13729:1999, Health informatics – Secure User Identification for Healthcare Strong Authentication Using Microprocessor Cards, 1999.
[14] CEN TC 251 N99-001, Draft CEN Report on Safety and Security Related Software Quality Standards for Healthcare (SSQS), 1999.
[15] Potamias G, and the INFOBIOMED consortium partners. “State of the Art on Systems for Data Analysis, Information Retrieval and Decision Support, Deliverable D13 v1.1 Final”, 23/01/2006.
[16] ISO/ TC 215, Health Informatics, Privilege Management and Access Control -- Part 1: Overview and Policy Management, Under Development Standard ISO/CD TS 22600-1, October 5, 2004.
[17] ISO/ TC 215, Health Informatics, Privilege Management and Access Control -- Part 2: Formal models, Under Development Standard ISO/CD TS 22600-2, August 16, 2005.
[18] ISO/ TC 215, Health Informatics, Trusted end-to-end information flows, Published Standard ISO/TR 21089:2004, June 15, 2004.
[19] http://intercare.imsgrp.net/
[20] http://www.hygeianet.gr/
[21] http://www.medcom.dk/picnic/
[22] http://coras.sourceforge.net
[23] http://www.infobiomed.org/

8. Conclusions

In the framework of the work-package WP3 - MIDDLEWARE, INTEROPERABILITY AND INTEGRATION of the HEARTFAID project, this document has dealt with the analysis of the functional specifications of what will be the supporting infrastructure of the HEARTFAID Platform of services (HFP), that is the Middleware. We analysed the main needs of the context in which the platform will be implemented, the weakness of the specific cardiovascular field, and finally the technological requirements that the middleware will have to meet in order to effectively support the achievements of the main and most ambitious project goals. We have also analysed which are the methodological foundations on which the future activities of the project will be based and we introduced some preliminary architectural specifications of the middleware infrastructure, together with an analysis of the most commonly used standards for clinical data acquisition, encoding and transmission, and the security issues that should be faced and that have a fundamental importance due to both the context in which the platform will operate and the sensitivity of the data that will be managed.

In more details, we have analysed the main aspects that actually characterise the heterogeneous world of the healthcare assistance, where we register a proliferation of a large quantity of non-interoperable ICT solution that are not able to communicate each with the other or to exchange data. We presented the most advanced solutions emerging in the ICT context, which are able to guarantee a better integration not only between the new developed tools but also between the new and the existing solutions.

After having reported the main aspects and needs of the healthcare system, we analysed in details the interoperability issues that are being faced in the eHelath domain. These issues will be the main challenges in the near future, not only in the cardiovascular field. We also analysed the profiles of the end-users of the HF platform, and the user needs, as well as the most common clinical standards that are being developed to increase the interoperable capabilities of the modern ICT tools.

The HF middleware will incorporate the relevant clinical standard and when feasible will translate clinical data received by medical devices in non-standard form in an internal standard format. Moreover all the issues about security will be strongly observed and implemented according to the relevant existing standards.

Concerning the communication with the sensors/medical devices, the main role of the middleware will be to provide to the applications an abstract way of communicating with the devices by taking away the complexity and the details of low-level communication with these sensors/devices. Furthermore, the middleware could provide data enhancement (e.g. time stamping) and data transformations (e.g. conversion of data to XML structures).

Regarding the transmission of data to the HEARTFAID platform, the middleware handles the communication details with the platform and reports possible problems to the end user application.

Regarding the design view of the HF RDA, when the data is acquired at the user's home or at the hospital, a PC will collect, eventually enhance and send the data to the HAEARTFAID platform in the form of XML messages, by using a simple Internet connection. However, when the data is collected on the move, a mobile phone can collect the data and send it to the HEARTFAID platform over IP using the wireless cellular network.

The middleware should support also the interaction with a decision support system (DSS), able to apply a reasoning process to a specific question. The DSS, in fact, should be connected to the KB and the End-user levels through he middleware that in its turn should support the management of the business processes.

In this way, the functional specification of the middleware should be therefore strictly related to the requirements of the DSS.

A controller that handles data and processes flows could be thought in different ways, as a workflow management or an ad-hoc system. When the flow does not change then the workflow solution is not fundamental. Otherwise, its advantage is that it is independent of the flow and if the latter evolves it is not necessary to rewrite the application that manages it, but we only need to redefine the flow using the process definition language supported by the workflow management system.

Finally, we presented a preliminary design view of the middleware and, in particular, of the communication infrastructure, the remote data acquisition network and the end-user applications.